A REVISTA DOS USUÁRIOS DO TK

MICROHOBBY

ANO I - Nº. 2 - Junho 83

ANTI-MÍSSIL 1K

Dicas
Novidades
Fita do Mês: SICOM
Alta Resolução
TK Poeta
Quebra-Cabeça: Verdadeiro ou Falso?

REVENDEDORES: ARACAJÚ 224-1310 • BELEM 222-5122/226-0518 • BELO HORIZONTE 226-6336/225-3305/225-0644/201-7555 • BLUMENAU 22-1250 • BRASÍLIA 224-2777/225-4534/226-9201/226-4327/242-6344/242-5159 • BRUSQUE 55-0675 • CAMPINAS 32-3810/8-0822/32-4155/2-9930 • CAMPO GRANDE 383-6487/382-5332 • CARUARU 721-1273 • CUIABÁ 321-8119/321-7929 • CURITIBA 232-1750/224-6467/224-3422/243-1731/223-6944/233-8572/232-1196 • DIVINÓPOLIS 221-2942 • FLORIANÓPOLIS 23-1039 • FORTALEZA 226-4922/231-5249/231-0577/231-7013 • FREDERICO WESTPHALEN 344-1550 • GOIÂNIA 261-0333/224-0557 • IJUÍ 332-2740 • ITAJUBÁ 622-2088 • LINS 22-2428 • LONDRINA 22-4244/23-9674 • MACEIÓ 223-3979/221-6776 • MANAUS 237-1793 • MOGI DAS CRUZES 468-3779/208-6797 • MURIARÉ 721-1593 • NATAL 222-3212/231-1055 • NITERÓI 722-6791 • NOVO HAMBURGO 93-1922/93-3800 • PELOTAS 24-5139 • PORTO ALEGRE 26-8246/21-4189/24-1411/22-3151/24-0311/21-6109/24-7746 • PRESIDENTE PRUDENTE 22-2788 • RECIFE 241-4310/224-8777/224-3436/224-4327 • RESENDE 54-1664 • RIBEIRÃO PRETO 636-0586/634-4715/635-1195 • RIO DE JANEIRO 267-1093/252-2050/253-3395/264-0143/259-1516/232-5948/591-3297/222-6088/267-1339/329-4869/228-2650/246-4824/239-5612/542-3849/62-8737 • SALVADOR 248-6666/235-4184/247-5717 • SANTA MARIA 221-7120 • SANTO ANDRÉ 455-4962/444-7375/454-9283 • SANTOS 4-1220/32-7045/35-1792/33-2230 • SÃO CARLOS 71-9424 • SÃO JOÃO DA BOA VISTA 22-3336 • SÃO JOSÉ DOS CAMPOS 22-3968/22-7311/22-8925/21-3135 • SÃO PAULO 853-0164/853-0448/239-4122/36-6961/61-4049/881-1149/258-3954/212-9004/282-2105/212-3888/545-4769/227-3022/864-8200/222-1511/259-2600/282-6609/813-4555/814-3663/826-1499/521-3779/270-7442/210-7681/813-4031 • SOROCABA 32-9988 • TAUBATÉ 31-4137 • UBERABA 333-1091 • UBERLÂNDIA 234-8796 • VIÇOSA 891-1790/891-2258 • MARÍLIA 33-4109

Expediente

PUBLICAÇÃO:
Micromega Publicações e Material Didático — Cx. Postal 60081
CEP 05096 — São Paulo — SP
DIRETOR:
Pierluigi Piazzi
JORNALISTA RESPONSÁVEL:
Aristides Ribas de Andrade Fº
ASSESSOR TÉCNICO:
Flavio Rossini
DIRETOR DE ARTE:
Fernando Leoni
ILUSTRAÇÕES:
Fatima M. Rossini Gouveia
COLABORARAM NESTE NÚMERO:
Nilson D. Martello, Bernardo Cláudio Stein, Igor Sartoni, Dong
CORRESPONDENTES:
Londres — Robert L. Lloyd
Paris — Alain Richard
N. York — Natan Portnoy
Milão — Licia Zappa
PERIODICIDADE:
Mensal
TIRAGEM:
30.000 exemplares
NÚMERO 2:
Distribuição só para assinantes.

ASSINATURAS:
12 meses: Cr$ 11.800,00
Promoção especial até 30 de setembro: Cr$ 9.900,00 com um brinde à sua escolha: uma fita com programa ou um desgravador instantâneo de fita.
DEPARTAMENTO COMERCIAL:
Tel.: (011) 257-5767
c/ srta. Rosana

Índice

CAPA: Leoni

Editorial

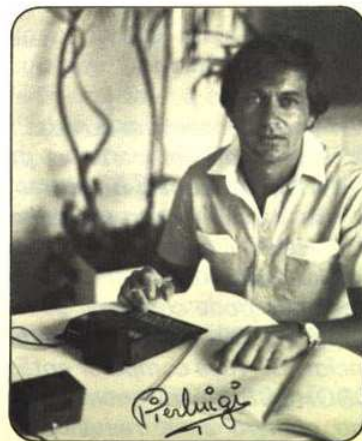
É extremamente embaraçoso começar um editorial com um pedido de desculpas, mas ele se faz necessário por dois motivos. Inicialmente o atraso do número 1 em relação ao prazo previsto por ocasião do lançamento do número Ø.

Uma série de dificuldades burocráticas aparentemente irrelevantes nos fizeram passar por uma situação típica dos pesadelos: correr, correr, sem sair do lugar.

Felizmente já entramos no ritmo normal e a periodicidade mensal será mantida, com o máximo de pontualidade.

Em segundo lugar, o atraso da fita brinde. Algumas dificuldades na produção nos fizeram optar por uma maior demora em troca de uma melhor qualidade no produto a ser enviado para nossos assinantes.

Além das desculpas, com este número estamos atendendo a uma série de solicitações dos nossos leitores, a principal delas referente ao "volume" da revista, cuja tendência daqui para frente é crescer cada vez mais. Começamos neste número um curso de linguagem de máquina que permitirá, a partir do próximo número, a publicação de programas especiais nesta linguagem.

Finalmente, queremos agradecer a confiança que uma multidão de novos assinantes depositou em nós, permitindo assim a elaboração de uma revista que desejamos cada vez melhor e cada vez mais útil.

CARTA DOS LEITORES

Prezados Senhores,

Achei interessante a idéia de V.Sas., em lançar uma revista voltada exclusivamente para o TK-82. Espero que saibam levar a publicação a um bom caminho e fujam de "bater na mesma tecla", muito comum em outras revistas do gênero. Boa sorte então!

Sobre a assinatura, tenho algumas dúvidas. Sou possuidor de um TK, com expansão para 16K, que me ajuda muito no curso que estou terminando de Engenharia Eletrônica. Não uso muito seus recursos de software. Procuro aprender mais e aplicar o que aprendi nas áreas de software básico e hardware. Comparando o preço da assinatura pretendido por V.Sas., com o preço de outras publicações, percebe-se boa diferença. Talvez devido ao 'brinde'. Não estou interessado na fita com programas. A assinatura sai no mesmo preço? Fica como sugestão o brinde opcional. Outra dúvida: O mercado de micros está em constante mudança. Hoje já vem o TK85 para substituir o TK82-C. Em breve, virão melhores micros. E a revista? Como fica? Acompanha a evolução, ou fica fiel aos antigos possuidores do TK? Talvez as duas coisas, mas até quando?

Luis Fernando Neves de Pinho

Prezado Luis,

Vamos esclarecer suas dúvidas:

1) É impossível comparar o preço da assinatura de MICROHOBBY com o de outras publicações do gênero, pois não existem outras publicações assim no Brasil. MICROHOBBY é uma revista para usuários do TK, mantida pelas assinaturas destes mesmos usuários.

Você não pode comparar seu custo com o de revistas que vendem 30%, 40% ou até 50% de seu espaço para anúncios. Se você comparar então o grau de utilidade de MICROHOBBY para quem tem um TK, você verá que só no número zero, reduzido e promocional há mais coisas para o TK do que em todas as outras revistas juntas.

2) O brinde é um presente nosso e não altera o preço da assinatura. Consequentemente, o aconselhamos a passá-lo para algum amigo que ficará muito agradecido.

3) O TK85 não veio para substituir o TK82 pois situa-se em outra faixa de mercado. Como eles usam a mesma lógica, seus softwares são compatíveis, portanto a revista é escrita para os usuários dos dois TKs e de todos os eventuais TK que venham a surgir.

4) Se a revista sentir a necessidade de atender outros micros certamente continuará também fiel aos possuidores do TK82 pois foi para eles que ela foi criada. Este é um compromisso que assumimos e vamos manter. Já recebemos por exemplo, solicitações para abrir um suplemento que atende computadores com lógica APPLE. A idéia está sendo estudada, mas fique tranquilo com uma coisa: qualquer outro computador que seja atendido por MICROHOBBY será algo A MAIS que o usuário do TK estará recebendo, sem roubar nenhum espaço a ele dedicado.

Prezados Senhores,

Recebi o número Ø e fiquei impressionado com o ótimo nível da revista. Ela está muito bem bolada. Faltava algo que unisse todos os possuidores do TK e similares. Vocês acertaram na mosca.

Eu já possuia contato com programação pois possuo uma calculadora HP-41CV desde o começo do ano passado. Em outubro me motivei a comprar um micro e optei pelo TK, pela simplicidade e preço reduzido. Aprender a linguagem BASIC não foi muito difícil (é uma questão de tempo). Não possuo por enquanto nenhum tipo de acessório. Possuo uma série de programas (principalmente jogos) feitos por mim. Muitas modificações podem ser feitas para melhorar os programas. Pretendo remeter-lhes mensalmente um bom programa ou uma colaboração. Inicialmente mando este JOGO DA FORCA que ao meu ver ficou muito melhor do que muitos (de 2K) que já vi. Contem com meus sinceros votos de felicidade e BOA SORTE.

Armando de Toledo Filho

Caro Armando,

Antes de mais nada, obrigado pelos votos de BOA SORTE e pela sua oferta de colaboração (veja matéria "Como colaborar com MICROHOBBY" na pág. 38). Com relação ao jogo de forca, ele foi analisado pelo pessoal da redação e você está de parabéns: o programa é ótimo. Infelizmente não vamos publicá-lo pois quase todas as outras revistas do ramo já publicaram vários "JOGO DA FORCA", "SENHA" e similares. Apesar disto, seu programa pode ser aproveitado: porque não inventa ou aperfeiçoa mais alguns jogos e produz uma fita de jogos para 2K? Você poderia até vender esta fita para interessados. Para isto abrimos um espaço gratuito de pequenos anúncios (veja pág. 39). Desejamos uma boa sorte para você também e continue enviando colaborações.

→

CARTA DOS LEITORES

Continuação...

Prezados senhores,

Possuo um TK82-C há pouco tempo. Entre os programas que comprei já prontos, um deles (MONSTRO DAS TREVAS) não carrega e se perde ao chegar no final. Gostaria de saber se existe algum "macete" para carregar programas como este, que aparentemente estão com defeito.

Sérgio Luis de Oliveira

Prezado Sérgio,

Quando isto acontecer tente 3 coisas nesta ordem:
1) Limpe a cabeça de gravação do seu gravador antes de tentar o LOAD, ajustando o controle de volume e tonalidade, até obter um som o mais estridente possível. Teste antes com um programa mais curto que entre e ouça o som produzido, tentando depois reproduzir a mesma altura e timbre no programa que lhe interessa.
2) Tente utilizar outro gravador com os mesmos cuidados do item 1.
3) Se os itens 1 e 2 não funcionaram, provavelmente a fita está com defeito. Envie-a para a MICROSOFT, com uma carta explicando o problema e o pessoal mandará para você uma fita novinha.

Prezados Senhores,

Muito oportuno o lançamento de uma revista dedicada ao TK. Já se tornava cansativo comprar as revistas especializadas com "dicas quentes" sobre dezenas de microcomputadores, exceto o TK-82.

Agora que vocês se anteciparam nesse negócio, aqui vão algumas sugestões:
a) Manter a mesma linha de todas as revistas do gênero, ou seja, divulgar programinhas sem graça. Isso satisfaz a maioria.
b) Promover os programas "quentes" mesmo que isso implique em gastar mais papel.
c) Divulgar, com espírito de formação, conhecimentos básicos, tais como os conceitos de "variável" e aritmética com números hexadecimais.
d) Fornecer dicas realmente quentes de como, por exemplo, movimentar desenhos ou informações no vídeo sem usar o comando PRINT, mas definindo diversas áreas dentro do programa BASIC e dando vários POKES nos endereços do DFILE (16396 e 16397).

Finalizando a presente, gostaria de manifestar o meu interesse em colaborar com vocês, pois, há muito desde 1968, emprego computadores em meus serviços e creio que na experiência obtida tenha muitas idéias que serão úteis para seus leitores.

Roberto Massaru Watanabe

Caro Roberto,

Agradecemos suas sugestões e seu interesse em colaborar com nossa revista. Esperamos ter, nos próximos números muitos artigos e programas com sua assinatura (veja "Como colaborar com MICROHOBBY", pág.38).

Com relação às suas sugestões, alguns comentários:
A) Vamos realmente publicar programinhas "fáceis" (que você chamou de "sem graça") pois uma considerável fração de nossos assinantes é constituída de crianças e principiantes. Muitos não tem expansão de memória, mas têm o direito de se divertir aprendendo. Num curso para crianças de 8 a 10 anos, realizado no NÚCLEO DE ORIENTAÇÃO DE ESTUDOS, o programa BOLICHE, publicado no número Ø, provocou muito entusiasmo. As crianças descobriram **sozinhas** *que alterando o "LOOP"*

```
155 FOR M = 1 TO 8
```

para:

```
155 FOR M = 1 TO 2Ø
```

passavam a ter direito a mais bolas!

Também sozinhas descobriram que eliminando as linhas:

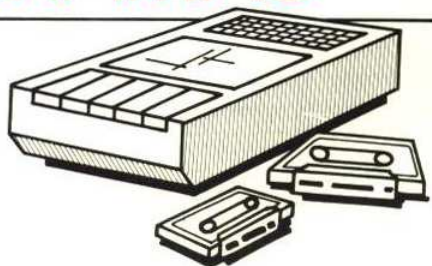
```
195 UNPLOT A, B
199 UNPLOT A, B
```

o programa passava a "desenhar foguetes"! Foi muito gratificante observar o resultado de um programinha "fácil" publicado na nossa revista.
B) Vamos gastar muito, mas muito papel justamente para publicar os programas "quentes". Se o leitor passar todos os programas de 2 números de nossa revista para fitas, terá uma quantidade de material cujo preço, no mercado, ultrapassa de muito o preço de uma assinatura anual (12 números).
C) A filosofia básica que norteia nossa publicação é muito mais "didática" do que "jornalística". É um dos nossos principais objetivos imprimir a esta publicação o "espírito de formação" que você propõe em sua carta. Como você pode notar, a partir deste número, além de continuarmos nosso curso de BASIC-TK para principiantes, estamos iniciando um curso de ASSEMBLY para o TK.
D) As dicas "quentes" já estão começando a surgir. Por coincidência, neste número temos uma dica que esclarece justamente um dos usos da variável DFILE que você cita em sua carta.

Muito mais dicas serão fornecidas nos próximos números mostrando aos leitores as inúmeras e insuspeitas possibilidades desta pequena maravilha que é o seu TK.

●

A GRAVAÇÃO DOS PROGRAMAS

Nilson D. Martello

Após muitos anos criticando os aparelhos de alta-fidelidade em algumas revistas brasileiras e enfrentando preconceitos, "rumores" e lendas fantásticas sobre Som e o "software" que o acompanha, chego a este novo campo dos computadores pessoais.

Leigo na matéria, alfabetizando-me em Basic, levando "surras" homéricas do meu TK, também começo a encontrar pessoas interessadíssimas; atendimento cuidadoso e gentil por parte de vendedores de lojas, representantes da Microdigital; de firmas particulares que comerciam micros ou programas, etc, a ponto de me perguntar:

— Ué?! O que está acontecendo?

De outro lado, também nesta área começo a encontrar os rumores e lendas fantásticas; exemplo . . . "Um gravador cassete, quanto mais barato melhor para gravar seu programa".

Daí a me atrever — eu, novato em **micro** — a trazer alguns princípios básicos aprendidos durante os anos em eletrônica de Som, para esta área.

O gravador

Não é verdade que o gravador "mais **barato**" seja o melhor. E nem poderia ser. O "guincho" gravado no cassete — as informações do **micro** — é rico em sons médio-agudos. Os próprios fabricantes insistem que, em tendo o aparelho um controle de tonalidade, seja colocado na posição de **mais agudo.** E é aqui que os micros saem-se mal: gravadores de baixa qualidade têm sua resposta de agudos muito pobres. O mesmo se dá com bons gravadores, usados (geralmente monoaurais) há longo tempo, que o usuário consegue comprar por bom preço. Estes gravadores já terão rodado muitos quilômetros de fita, desgastando os cabeçotes de gravação/reprodução, o que influencia precisamente mais na gama dos médio-agudos.

Por outro lado, gravadores "mais baratos" possuem limitações mecânicas que logo conduzem à lata de lixo. Se as flutuações desses sistemas mais pobres tornam tais gravadores imprestáveis para a reprodução de música, o mesmo não é tão grave quanto aos sinais digitais. Uma certa flutuação (**wow** e **flutter**) ainda é suportada galhardamente nos momentos de **LOAD** e **SAVE**. Porém as funções de rebobinamento rápido (para a esquerda ou direita), além da possível interrupção do movimento da fita com cassetes semi-defeituosas, tornam o aparelho de baixa confiabilidade, isto é, dor de cabeça para seu usuário.

Ainda mais, os aparelhos de baixo preço no geral possuem marcas estranhíssimas, como "Claudicorder", "Somissuko" ou "Mankassom", fabricados na Konxinchina, Taboão da Serra ou Trás-dos-Montes. Quando chega o momento de consertá-lo e substituir um componente defeituoso, o técnico será obrigado a "improvisar". Resumo: mais dinheiro jogado fora.

Por fim, aparelhos baratos não possuem contador numérico do andamento da fita. Para quem não sabe, tais contadores são de baixa precisão e, geralmente, contam as voltas da bobina, dentro do cassete. Por isso raras vezes a leitura é válida entre aparelhos diferentes. De qualquer forma, quando gravamos diversos programas em uma mesma fita cassete, torna-se quase impossível localizar o programa específico sem o auxílio dos numerozinhos.

Sugestões:

1) Compre um gravador de marca conhecida e **respeitada,** ainda que não esteja em condições 100%. Uma vez que o aparelho tenha representante no país — ou seja aqui fabricado por indústria de respeito, que mantenha eficiente serviço de manutenção — tudo bem.

2) Mande examiná-lo antes de começar a empregá-lo. O mínimo que se pede é uma boa limpeza e lubrificação, com desmagnetização dos cabeçotes e guias metálicas, assim como substituição dos potenciômetros (controles de volumes, de tom) defeituosos ou ruidosos. Aliás, manda o mínimo de cuidado que **ao menos** uma vez por mês os cabeçotes sejam limpos com álcool metílico e desmagnetizados. Se os cabeçotes apresentarem desgastes, mande substituí-los. Não custam assim tão caros e prestarão serviço por longo tempo, se bem cuidados. Lembre-se: cabeçotes desgastados gravam e reproduzem muito mal os sons agudos.

3) Prefira uma **boa** fonte de alimentação (ou conversor AC/DC, ou Adaptador); o preço das baterias não anda brincadeira e, a menos que você empregue baterias alcalinas ou de mercúrio, verá o gravador "arrastando a fita com a língua de fora", após pouco tempo de uso.

A GRAVAÇÃO DOS PROGRAMAS

Continuação...

A fita cassete

Não vamos citar nomes para evitar polêmicas desnecessárias. Porém algumas fitas nacionais — mesmo de certo renome e preço — desprendem o material magnetizável que passa a sujar os cabeçotes. A reprodução que mais sofre são os médio-agudos, com frequência insuportável. Cabe aqui a consideração feita em reprodução de alta-fidelidade: "Se vale a pena ser gravado, vale a pena ser BEM gravado".

Empregue fitas de renome, de baixa duração (nunca as de 90 ou 120 minutos) e preserve-as com carinho. Nada de proximidades de fontes magnetizadoras (transformadores, cinescópio de TV, etc); no mínimo a 30 cm de distância, o que assegurará que os sinais gravados não se alterarão. O calor, a umidade e a poeira (porta-luvas de carro, por exemplo) são os próximos inimigos a serem evitados. Não se esqueça de que o próprio **micro** costuma aquecer; nada de espantar, é claro, porém não comece a testar o ponto de fusão do plástico empregado na cassete! Da mesma maneira, não toque com os dedos na fita; você não sabe como nossa pele é gordurosa! Se houver algum acidente que o obrigue a desmontar o cassete, ou emendar a fita, manuseie-a com luvas de algodão. A fita e os cabeçotes agradecerão o cuidado.

Nível de gravação

Os indicadores de nível de gravação — ou VU-meter — são um auxiliar precário no julgamento do nível de gravação. A última palavra será dada pelo seu **micro** e a imagem reproduzida no monitor ou TV falarão mais "alto". . . A limitação dos VU-meter explica-se facilmente: são um tanto "totalizadores", ignorando se o sinal mais intenso está na gama dos médios, agudos ou graves. Isto é. . . isso na melhor das hipóteses, quando foram calibrados na fábrica. Comparando equipamento similar, e, por vezes, até modelos idênticos, nacionais ou importados, já se pode perceber uma total discrepância na indicação dos diferentes VU-meters. É claro que não deveria acontecer, mas acontece . . .

Pelo mesmo motivo, as gravações em "Automático" nem sempre funcionam. Dependendo do circuito eletrônico empregado num específico gravador, da fita empregada, e finalmente das **neuroses** do seu **micro**, o **LOAD** poderá parecer a contento numa situação e inútil em outra. O mais prático é digitar programas curtos e testar os melhores níveis de gravação (**SAVE**) e reprodução (**LOAD**).

Conectores

Já presenciamos um profissional **rosnar** contra o cabo conector entre o computador e o gravador cassete:

— Está com mau contato!

Se há tal desconfiança, leve o cabo a qualquer técnico de eletrônica para ser consertado; não deixe as coisas assim! Ou compre um par de cabos extras, para não ficar parado à espera do conserto. De todos os **soft,** os cabos são de longe os mais baratos.

Em alta-fidelidade, em situações quando trabalhamos com sinais minúsculos, empregamos até contatos banhados a ouro para os conectores. Isto não se aplica aos micros. No entanto a boa qualidade dos contatos é imprescindível para se evitar oxidação, principalmente para aqueles que vivem à beira-mar ou em locais de alta umidade relativa. Gaste um pouco mais num bom cabo com conectores bem acabados e evitará as incertezas de um mau contato intermitente.

TV à Cores

"O **micro** estraga o TV à cores" — é outra falácia que já ouvi algumas vezes, em locais diferentes.

O sintonizador de seu TV não tem a menor idéia se está recebendo um sinal direto do transmissor de TV do canal 2, ou do gerador de seu vídeo-cassete, ou de seu **micro** (supondo que essas fontes estejam funcionando corretamente, é claro). O que salta aos olhos — ou aos bolsos, melhor dito — é que, por menos tempo que você empregue o computador, estará desgastando o tubo de imagem do televisor. Ora, o componente mais caro é precisamente o tubo de imagens. Em se tratando de um modelo à cores, custará de três a quatro vezes o preço de um tubo branco e preto de tamanho similar. A mão-de-obra para a troca é proporcionalmente mais cara também. Além disso, imagine se, por azar, coincide de o aparelho mostrar defeito na hora da telenovela precisamente **logo depois** que você acabou de empregá-lo com o **micro**?

— E as Crianças?

— Quais crianças? — perguntará o leitor, já alarmado.

Pois é. . . as crianças da casa — filhos, irmãos menores, priminhos de fins de semana — simplesmente se encantam com o **micro.** E a ordem: "Não ponha a mão aí!" . . . além de antipática, seria ineficiente. Aliás, mais do que antipática, estaríamos sendo anti-pedagógicos. Dependendo da idade da criança, podemos ensinar desde o manuseio de simples jogos até princípios de manipulação do teclado. É simplesmente inacreditável a velocidade de aprendizado de crianças de seis, sete ou oito anos. Também é imperioso que nos convençamos de que o **micro** veio para ficar **também** na vida das crianças. Se gastarmos duas horas em cima de um programa dedicado a nossos filhos ou irmãos menores, propondo-lhes problemas de aritmética, história ou conhecimentos gerais, poderemos estar empregando as melhores duas horas de nossa vida.

Mas isso já é outra conversa que pouco tem a ver com o Som para os **Micros.** Ficam aí as sugestões que, espero, sejam de utilidades para você. ●

IF... THEN... ELSE para o TK

Igor Sartoni

Quantas vezes, você viu um programa interessante e não pode rodá-lo em seu computador, por um problema ou outro? Quantas vezes, isso não foi possível, porque, apesar do programa estar escrito em BASIC, ele possui as instruções que não podiam ser usadas no TK?

Isso ocorre, porque, embora o BASIC tenha se tornado praticamente um padrão como linguagem de computação em microcomputadores pessoais, podemos encontrar certas peculiaridades inerentes a cada modelo, o que torna, muitas vezes, difícil transpor um programa de um computador para outro.

Certas instruções aparecem com outra sintaxe e o que é aceito por um computador, não é aceito por outro. Outras surgem em um e não no outro, como é o nosso caso: o TK-82 possui a instrução **IF . . . THEN,** mas não possui seu complemento, a condicional **ELSE.**

IF . . . THEN . . . ELSE

A instrução **IF** (condição) **THEN** (função ou operação) é comum a todos os computadores que usam BASIC. No entanto, apenas alguns possuem a condicional **ELSE** (função ou operação).

Esta condição diz que, se a condição que segue a condicional **THEN** não for satisfeita, uma outra operação ou função deve ser executada. Um exemplo pode deixar mais claro. Observe a linha de programa a seguir:

```
100 IF X = 0 THEN LET Y = 0 ELSE
PRINT "X="; X
```

Nesta linha de programa, **se** (**IF**) a primeira condição for satisfeita (X = Ø), então (**THEN**) realizaremos a operação que lhe segue (**LET** Y = Ø), senão (**ELSE**), realizaremos a outra operação (**PRINT** "X="; X).

Computadores como o TK-82 ou o TK-85 não possuem esta condicional. Como incluí-la no seu computador? Examinemos o que ela faz. Pela descrição que fizemos anteriormente, vimos que ela estabelece uma condição oposta àquela determinada pela condicional **IF**. A primeira idéia que nos vem à mente é transformar a condicional **ELSE** numa linha de programa que expresse uma condição oposta à determinada pela condicional **IF**. No exemplo que mostramos, a linha de programa seria transformada em:

```
100 IF X = 0 THEN LET Y = 0
110 IF X <> 0 THEN PRINT "X="; X
```

A solução foi facilmente estabelecida, porque a condição estabelecida na condicional **IF** era a igualdade, cujo oposto é a desigualdade (< >). E se a condição fosse outra, como, por exemplo, maior ou igual a (>=)? Neste caso temos duas opções. A primeira delas é usar a função lógica **NOT**. Esta função age como um inversor, tornando a lógica da condição invertida: se antes era verdadeira, ela torna-se falsa e vice-versa. Deste modo a linha:

```
100 IF X >= 0 THEN LET Y = 0 ELSE
PRINT "X="; X
```

Poderia ser traduzida como:

```
100 IF X >= 0 THEN LET Y = 0
110 IF NOT X >= 0 THEN PRINT "X="; X
```

A segunda opção, é usar o sinal complementar do sinal da condição. Para entendermos o que isso significa, vamos escolher uma sequência de números naturais em ordem crescente, como, por exemplo:

Ø 1 2 3 4 5 6 7 8 9

Poderemos separar uma parte desta sequência, aplicando uma condição a um dos seus elementos. Por exemplo, se desejarmos uma sequência formada por:

6 7 8 9

Poderemos dizer que tomamos todos os números maiores ou iguais a seis, ou seja, aplicamos a condição "maior ou igual a" é a condição que fornece a parte desprezada da sequência. No nosso exemplo, são os números menores que 6. Assim, a condição "menor que" é a complementar a "maior ou igual a".

A tabela a seguir fornece todas as condições com seu respectivo símbolo e complemento:

Condição	Símbolo	Complemento	Símbolo
maior que	>	menor ou igual a	<=
menor que	<	maior ou igual a	>=
maior ou igual a	>=	menor que	<
menor ou igual a	<=	maior que	>
igual a	=	diferente de	<>
diferente de	<>	igual a	=

Assim, a linha do exemplo ficaria:

IF... THEN... ELSE para o TK

Continuação...

```
100 IF X>=0 THEN LET Y=0
110 IF X<0 THEN PRINT "X=";X
```

Estas duas soluções resolveram o problema. Entretanto, existem mais duas soluções. Estas soluções, embora necessitem um número de linhas maior que as anteriores, evitam o aparecimento de duas condicionais em seguida, usando uma característica da própria instrução **IF/THEN**. Nesta instrução, quando a condição não é satisfeita, o programa executa a linha que vem imediatamente a seguir. E se colocarmos nesta linha a função ou operação designada pela condicional **ELSE**? Isto não seria uma boa solução porque, se fizermos apenas esta modificação, a função que segue o **ELSE** será executada em qualquer condição, mas o que desejamos é que ela seja executada se, e somente se, a condição estabelecida pela condicional **IF** não for satisfeita. Isto pode ser evitado se "saltarmos" esta condição, por meio de uma instrução **GOTO**. Assim, nosso exemplo ficará:

```
100 IF X=0 THEN GOTO 130
110 PRINT "X= ";X
120 GOTO 140
130 LET Y=0
140 ... (sequência do programa)
```

A linha 120 é necessária pelo mesmo motivo que usamos no salto da linha 100: desejamos que a função ou operação designada pela instrução **IF** original, seja executada apenas se a condição for satisfeita. Se ela for omitida isto não ocorre.

A outra solução é semelhante a esta, só que usa uma lógica invertida para fazer a comparação, ou seja, é feita primeiro a comparação para se realizar a função designada pela condicional **ELSE**. Nosso exemplo ficaria assim:

```
100 IF X<>0 THEN GOTO 130
110 LET Y=0
120 GOTO 140
130 PRINT "X= ";X
140 ... (sequência de programa)
```

Outras Soluções

As soluções apresentadas neste artigo são de caráter genérico e devem funcionar em quaisquer circunstâncias, respeitadas as limitações de seu micro. Todavia, se você dispuser do diagrama em blocos do programa original que você deseja transpor para o TK, você pode, com um pouco de habilidade, modificar sua lógica de modo a evitar o uso da instrução **IF ... THEN ... ELSE ...**

A desvantagem de uma solução obtida desta forma é que ela está vinculada a um caso particular e só pode ser obtida se você tiver em mãos o diagrama em blocos, ou se você conhecer suficientemente a linguagem BASIC, a ponto de conseguir fazer esta análise sem auxílio do diagrama de blocos.

As outras soluções, por outro lado, só dependem de você localizar onde aparece a condicional **ELSE** e aplicar uma das soluções propostas. Só surgirão problemas, se vários **IF ... THEN ... ELSE ...** estiverem presentes na mesma linha. Mas isso será assunto de um próximo artigo.

Programa Exemplo

Não convém ficar só na teoria, tente verificar o que você aprendeu com o auxílio deste exemplo. Suponha que você tem diante de si o seguinte programa, feito para um computador, cuja única diferença entre ele e o seu, seja a presença da instrução **IF/THEN/ELSE**:

```
10 REM EQUAÇÃO DE SEGUNDO GRAU
20 PRINT "A= ";A
30 INPUT A
40 PRINT "B= ";
50 INPUT B
60 PRINT "C= ";
70 INPUT C
80 LET DELTA= "B**2-4*A*C
90 REM A PRÓXIMA INSTRUÇÃO CONTÉM A CONDICIONAL ELSE
100 IF DELTA >= 0 THEN LET D=SQR (DELTA) ELSE STOP
110 LET X1=(-1*B+D)/(2*A)
120 LET X2=(-1*B-D)/(2*A)
130 PRINT "AS RAÍZES DA EQUAÇÃO SÃO:"
140 PRINT "X1= ", X1, "X2= ", X2
```

Certamente você deve conhecer outros algorítmos para resolver a equação de segundo grau por computador melhores do que este, mas o programa que acabamos de mostrar, serve apenas para ilustrar o que dissemos sobre a instrução **IF ... THEN ... ELSE ...**

Solução

Mostraremos apenas uma das soluções. As outras ficam por sua conta. A linha que você vai ter de modificar é a linha 100:

```
100 IF DELTA >= 0 THEN LET D=SQR (DELTA) ELSE STOP
```

Vamos dividí-la em duas: uma, que contém apenas da condicional **IF** e da função que segue ao **THEN**:

```
100 IF DELTA >= 0 THE LET D = SQR (DELTA)
```

IF... THEN... ELSE para o TK

Continuação...

Outra, formada por uma condicional que vai verificar uma condição oposta à da condicional **IF** da nova linha 1ØØ. Temos duas opções: a função **NOT** e a condição complementar. Usaremos a segunda opção. A condição complementar de "maior ou igual a", de acordo com a tabela, é "menor que". Assim, devemos acrescentar a seguinte linha:

1Ø5 **IF** DELTA < Ø **THEN STOP**

Rode o programa em seu computador, testando-o com vários valores de A, B e C. Verifique também, as outras soluções, vendo inclusive, se não existe uma outra maneira de se elaborar o programa sem recorrer à instrução **ELSE**. O diagrama de blocos pode ser facilmente levantado a partir da listagem, visto que é um programa bastante simples.

Conclusão

Com o que você aprendeu, tente verificar se alguns dos programas publicados em outras revistas para outros computadores podem, com auxílio destes "macetes", ser rodados no seu TK. Se surgirem dificuldades em relação a estas ou outras instruções, escreva-nos contando exatamente o que lhe aconteceu, citando o computador para o qual foi destinado o programa, a instrução que lhe causou problemas, e uma cópia da listagem do programa que você tentou "traduzir" para seu computador. Este material servirá como ponto de partida para futuros artigos e, desde já, lhe agradecemos por qualquer colaboração prestada.

DESGRILANDO "FLAGS"

Flavio Rossini

Temos recebido cartas de muitos leitores pedindo sugestões de como conseguir detectar erros em programas muito complexos e cheios de desvios, onde fica difícil seguir passo a passo o caminho feito pelo programa. Podemos responder estas perguntas com um exemplo bastante elucidativo:

imagine uma linha de trens bem intrincada com muitos desvios.

figura 1

Suponha que um trem irá passar por uma dada região e você gostaria de saber qual o caminho percorrido pelo trem. Mas, por algum motivo, você não poderá estar lá para ver o trem passar. Imagine então que para cada **desvio,** haja um mecanismo que levanta uma bandeira se o trem passar por lá. Assim, se inicialmente todas as bandeiras estiverem abaixadas, basta verificar QUAIS estão levantadas após o trem passar para saber o caminho percorrido pelo programa. Observe o diagrama de blocos genéricos onde temos 5 desvios.

figura2

DESGRILANDO "FLAGS"

Continuação...

No bloco de iniciação (INI), basta acrescentar FLAGS abaixadas, o que pode ser feito usando a convenção de Ø (zero) para FLAG abaixada e 1 (um) para levantada; desse modo, no bloco INI basta colocar:

```
DIM  F (1Ø)
FOR  I = 1 TO 1Ø
LET  F (I) = Ø
NEXT I
```

Ou seja, 1Ø FLAGS, duas para cada desvio.

Feito isso, basta colocar no **início** de cada bloco de saída dos desvios uma instrução para "levantar" a bandeira:

```
início do bloco 1 ⇒ LET F (1) = 1
início do bloco 2 ⇒ LET F (2) = 1
.
.
.
início do bloco 9 ⇒ LET F (9) = 1
início do bloco 1Ø⇒ LET F (1Ø) = 1
```

Após executar o programa, basta verificar o estado dos FLAGS:

```
FOR I = Ø TO 9
SCROLL
PRINT "FLAG" ; I ; "ɃɃ" ; F (I)
NEXT I
```

Isto deverá ajudá-lo a detectar alguns erros difíceis de se encontrar nos seus programas "intrincados".

. . . In another world we could stand on top of the mountain with our flag unforled . . .

(Paul McCartney, 1982)

●

FITA DO MÊS SICOM

O programa SICOM-16K é produzido pela MICROSOFT e pode ser rodado em qualquer computador com BASIC – TK dotado de 16K de RAM.

Quem tiver um TK82 com expansão 64K ou um TK85 de 48K, deve tomar cuidado: como o programa foi concebido para 16K e um excesso de memória disponível, cria um "grilo": o computador não aceita dados novos. A solução, enquanto a MICROSOFT não lançar um SICOM especial para estes casos, é enganar o computador. Para isto basta alterar o valor da variável do sistema RTP (RAM TOP) fazendo:

```
POKE 16389, 192        (NEW LINE)
NEW                    (NEW LINE)
```

Após este procedimento o computador "pensa" que tem apenas 32K de RAM e, ao ser carregado com o SICOM, passa a operar tranquilamente.

O SICOM é um programa versátil que pode ser visualizado como um fichário. Cada ficha já tem previamente definidos os seguintes campos:

NOME	64 bytes no máximo
ENDEREÇO	64 bytes no máximo
CÓDIGOS DE INTERESSE . . .	64 bytes no máximo
TEXTO ADICIONAL	64 bytes no máximo

(cada byte = 1 caractere incluindo espaços vazios)

A quantidade de fichas armazenáveis é em função do número de bytes consumidos em cada uma (lembrando ainda que cada ficha consome 5 bytes adicionais para separar os campos).

Vamos verificar então quantos bytes podem consumir uma ficha para uma aplicação típica, por exemplo, cadastro de clientes.

Inicialmente carregamos o SICOM no computador e digitamos

```
GOTO 1
```

Nunca devemos digitar **RUN**, pois há uma variável que é apagada da memória tornando impossível a operação do programa. Neste momento, aparecerá na tela

A FITA DO MÊS
SICOM

Continuação...

o "menu" principal do programa (fig. 1). É aconselhável tirar a fita do gravador e armazená-la em lugar seguro, ao abrigo de campos magnéticos, etiquetando-a como

"fita mestre avó"

```
SICOM
0=INSIRA    3=BUSCA     6=PROX. PAG
1=ALTERA    4=IMPRIME   7=PROX.ITEM
2=CANCELA   5=SAVE      8=FORMATO:F
                        9=ESTATIST.
```

Figura 1

Convém agora fazer uma cópia que denominaremos:

"fita mestre mãe"

Digitamos 5 e o computador pergunta que nome queremos dar ao fichário (fig. 2), mostrando em vídeo reverso, qual ítem do menu está sendo ativado (no caso o 5).

Vamos digitar MICROHOBBY (**NEW LINE**). A seguir data e hora. Digitamos

20/06/83   08:30

```
SICOM
0=INSIRA    3=BUSCA     6=PROX. PAG
1=ALTERA    4=IMPRIME   7=PROX.ITEM
2=CANCELA   5=SAVE      8=FORMATO:F
                        9=ESTATIST.



ENTRE NOME DO ARQUIVO
""
```

Figura 2

O computador manda agora a mensagem:

ENTRE NL SE FITA PRONTA

Colocamos então, o gravador na posição gravar (REC) e com a fita virgem já correndo, apertamos **NEW LINE**. Os riscos característicos do **SAVE** aparecerão na tela da TV. Após 1,5 minutos a fita mãe estará pronta. Convém guardá-la em lugar seguro e começar a operar as fichas.

Vamos agora introduzir algumas fichas de assinantes da revista no computador.

Digitando 9 temos o número de fichas de bytes livres (fig. 3):

FICHAS 0   BYTES LIVRES = 12495

```
MICROHOBBY
0=INSIRA    3=BUSCA     6=PROX. PAG
1=ALTERA    4=IMPRIME   7=PROX.ITEM
2=CANCELA   5=SAVE      8=FORMATO:F
                        9=ESTATIST.
FICHAS=0    BYTES LIVRES=12495
```

Figura 3

Digitando 8, podemos escolher o formato do "display" na tela da TV (fig. 4).

```
MICROHOBBY
0=INSIRA    3=BUSCA     6=PROX. PAG
1=ALTERA    4=IMPRIME   7=PROX.ITEM
2=CANCELA   5=SAVE      8=FORMATO:F
                        9=ESTATIST.
F=COMPL, L=ENDER, ETIQUETA
A=ENDER,   I=INT   T=RESUMO
```

Figura 4

F = COMPLETO (mostra todos os campos)
L = ETIQUETA (mostra apenas NOME e ENDEREÇO)

Tanto o formato F quanto o L mostram apenas uma ficha de cada vez na tela. Para exemplificar, vamos digitar duas fichas (nos formatos F e L, uma vírgula no meio do campo, produz a impressão do que a segue na linha seguinte). Vamos escolher o formato F e digitar

A FITA DO MÊS
SICOM

Continuação...

∅. A seguir, vamos introduzir os dados da primeira ficha (fig. 5).

```
0=INSIRA   3=BUSCA    6=PROX. PAG
1=ALTERA   4=IMPRIME  7=PROX.ITEM
2=CANCELA  5=SAVE     8=FORMATO:F
                      9=ESTATIST.

ANTONIO FERREIRA





ENTRE ENDER.
"R.MARGARIDA 47,09875 BOTUCATU C
E"
```

Figura 5

Como estamos no formato F a ficha aparece completa com nome e endereço do assinante, data de vencimento da assinatura e telefone (fig. 6). Note também o efeito

```
0=INSIRA   3=BUSCA    6=PROX. PAG
1=ALTERA   4=IMPRIME  7=PROX.ITEM
2=CANCELA  5=SAVE     8=FORMATO:F
                      9=ESTATIST.

ANTONIO FERREIRA
R.MARGARIDA 47,
09875 BOTUCATU CE
MAI084
543.8766
```

Figura 6

da vírgula no campo do endereço. Vamos agora introduzir uma segunda ficha (fig. 7). Digitando agora 9, temos idéia de quantas fichas estão no arquivo e quantos bytes livres ainda restam:

FICHAS = 2 BYTES LIVRES = 12347

```
0=INSIRA   3=BUSCA    6=PROX. PAG
1=ALTERA   4=IMPRIME  7=PROX.ITEM
2=CANCELA  5=SAVE     8=FORMATO:F
                      9=ESTATIST.

JOSE* SDRUWS
AV.DOS CORVOS 675 AP.71,
90001 INDAIATUBA GO
JUN84
78.8888
```

Figura 7

Um rápido cálculo, permite verificar que neste rítmo podemos armazenar ainda umas 170 fichas. Como cadastro, isto já é um número considerável, especialmente levando-se em conta a função de busca do programa.

No formato L as mesmas fichas teriam aspecto da fig. 8.

```
JOSE* SDRUWS
AV.DOS CORVOS 675 AP.71,
90001 INDAIATUBA GO
```

Figura 8

Já no formato A, I ou T a tela mostra uma ficha por linha: na primeira metade da linha, os primeiros 16 bytes do nome e na segunda metade, os primeiros 16 bytes do campo selecionado:

A = endereço
I = código de interesse
T = texto adicional

A fig. 9 mostra o display nos 3 casos.

Qualquer erro na inserção de uma ficha, pode ser corrigido pelas funções:

1 = ALTERA
2 = CANCELA

Para procurar uma ficha basta digitar 3. Aparecerá um menú secundário, solicitando o campo a ser pesquisado:

N = NOME A = ENDER.
T = TEXTO I = INTS

Digitando a letra correspondente, o programa pede para que você entre com os detalhes do campo em questão. Os campos N, A e T comparam o que você digitou

A FITA DO MÊS
SICOM

Continuação...

```
ANTONIO FERREIRA R.MARGARIDA 47,
JOSE* SDRUWS     AV.DOS CORVOS 6


ANTONIO FERREIRA MAIO84
JOSE* SDRUWS     JUN84


ANTONIO FERREIRA 543.8766
JOSE* SDRUWS     78.8888
```

Figura 9

como detalhe "AN", serão selecionadas todas as fichas cujo campo N comece por AN.

Se você está no formato F ou L aparecerá uma ficha por vez. Para ver a seguinte, basta digitar

6 = PRÓXIMA PÁGINA

No formato A, I ou T você verá o resumo de até 12 fichas. Digitando 6, você verá outra lista, digitando 7, você produzirá um efeito análogo ao do **SCROLL.**

A qualquer momento, você pode tirar uma cópia da tela, na impressora, usando a função 4.

A busca no campo I é a mais sofisticada. Ex.: ao pedir detalhes, o computador compara todos os caracteres digitados, com todos os caracteres do campo I, **independentemente de sua ordem.** Todas as fichas que contém os caracteres indicados na busca, são selecionadas. Note, que isto, permite classificar cada ficha em até 64 categorias diferentes.

Mas não é só em cadastro de nomes que o SICOM revela sua utilidade; ele pode ser usado em qualquer tipo de arquivo, como por exemplo, bibliografia, agenda telefônica, etc. Para informações esquemáticas até umas 500 fichas podem ser gerenciadas pelo SICOM, usando seu sistema de BUSCA.

Para outras finalidades, é conveniente mexer um pouco no programa. O título dos campos, por exemplo, pode ser alterado mexendo-se nas linhas 5Ø5, 515, 525, 535, 6Ø6 e 1315.

Por exemplo, para preparar fichas bibliográficas, estas linhas poderiam ser mudadas para o indicado na figura 1Ø.

```
505 LET P$="00AUTOR"
510 GOSUB RE
515 LET P$="01TITULO"
520 GOSUB RE
525 LET P$="01CODIGOS BUSCA"
530 GOSUB RE
535 LET P$="02ANO-EDITORA"

 606>PRINT "N=AUTOR A=OBRA I=CODIGOS T=EDIT."

1315 PRINT "F=FICHA COMPLETA L=AUTOR+OBRA    A=OBRA I=CODIGO T=ANO+EDITORA"
```

Figura 10

Convém alterar apenas os nomes dos campos sem alterar as letras de endereçamento para não ter que modificar muito profundamente o programa.

Uma vez atualizado o arquivo, ele deve ser gravado com a data e a hora (função 5) em uma fita virgem. A cada atualização o programa deve ser transferido para fita mas sem apagar o registro anterior. Desta forma qualquer "grilo" com o último programa implica na perda apenas das últimas atualizações.

Como este programa já está no mercado há um certo tempo, com certeza algum leitor já deve ter descoberto alguma utilização bem original ou algum aperfeiçoamento útil. Estamos à espera, portanto, de alguma colaboração neste sentido para podermos publicá-la num dos próximos números de MICROHOBBY. ●

PROGRAMA DO MÊS

ANTI-MÍSSIL

Muitos de nossos leitores possuem outros microcomputadores que usam uma lógica compatível com a do TK. Alguns deles, como os primeiros modelos do NEZ8000 da Prológica ou ZX81 da SINCLAIR têm apenas 1K de RAM. Para eles estamos publicando este programa que é uma verdadeira acrobacia em termos de economia de memória.

A digitação deste programa tem então duas utilidades: uma é o próprio programa em si, que gera um joguinho divertido (dentro, é óbvio, das limitações de memória).

Um míssil (V) desce segundo uma trajetória que sofre desvios aleatórios. Você comanda um anti-míssil terra-ar (+) que deve destruir o míssil. No fim do jogo aparece o número de mísseis destruídos.

Para guiar seu anti-míssil, você usa as teclas:

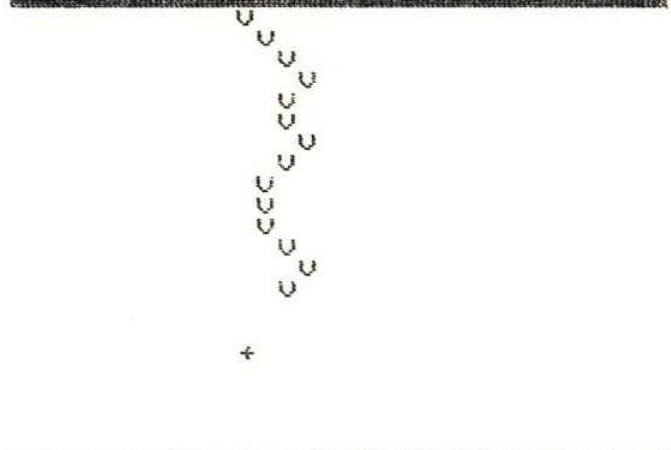
5
6
7
8

ou seu joystick.

A outra utilidade é o aprendizado de alguns "truques" para economizar memória.

Ao digitar

GOTO VAL "2Ø"

ANTI-MISSIL

```
   0▒REM MIROHOBBY 83
   1 FOR N=PI/PI TO VAL "9"
   2 LET A=VAL "20"
   3 LET B=VAL "11"
   4 LET X=PI-PI
   5 LET Y=B
   6 FOR F=PI-PI TO VAL "63"
   7 PLOT F,PI-PI
   8 NEXT F
   9 PRINT AT A,B;"+"
  10 PRINT AT X,Y;"V"
  13 PRINT AT A,B;" "
  15 LET Z=INT (RND*VAL "3")
  16 LET X=X+1
  17 LET Y=Y+(Z=VAL "1")-(Z=VAL
"0")
  18 IF INKEY$="6" THEN LET A=A+
2
  19 LET B=B+(INKEY$="8")-(INKEY
$="5")
  20 LET A=A-(INKEY$="7")
  21 IF A=X AND B=Y OR X>=22 THE
N GOTO VAL "23"
  22 GOTO VAL "9"
  23 CLS
  24 IF A=X THEN NEXT N
  25 IF X=VAL "22" THEN PRINT N
```

ao invés de

GOTO 2Ø

você está economizando 3 bytes!

Da mesma forma, ao fazer

X = PI / PI

ao invés de

X = 1

você livra 4 bytes.

Para se convencer disto, antes de digitar o ANTI-MÍSSIL, digite o seguinte programinha: →

PROGRAMA DO MÊS
ANTI-MÍSSIL

Continuação...

1 **GOTO** 2
2 **PRINT PEEK** 164Ø4 + 256 * **PEEK** 164Ø5 – 1651Ø

Ele indica, na tela, quantos bytes de memória o programa está ocupando. Anote o número indicado e mude a linha 1 para

1 **GOTO VAL** "2"

Compare o número de bytes ocupados agora com o valor obtido anteriormente.

Os primeiros modelos do NEZ8000 e do TK82-C não tinham **SLOW**. Por isso, quem tentou rodar o programa BARÃO VERMELHO do nº Ø num destes equipamentos deve ter notado que não funciona, a menos que se coloque uma instrução:

PAUSE 3Ø

após cada instrução **PRINT**.

No programa ANTI-MÍSSIL este cuidado também é indispensável se seu computador não tiver **SLOW**. Por isso acrescente a linha

12 **PAUSE** 3Ø

Haverá o inconveniente da tela ficar piscando, mas assim mesmo você poderá jogar e se divertir.

PROGRAMA DO MÊS
ALTA RESOLUÇÃO NA IMPRESSORA DO TK

Quando fazemos um gráfico na TV usando o

PRINT AT

estamos gerando um desenho, que tem amplas possibilidades nas nuances (pois podemos utilizar vários caracteres diferentes), mas que tem baixa resolução. Como os caracteres gráficos podem ser dispostos ao longo de 32 colunas e 22 linhas, (ou no máximo 24 usando as linhas de edição), temos uma matriz gráfica de no máximo

24 X 32

posições.

Já a instrução

PLOT

permite uma melhor definição gráfica, pois cada unidade por ela gerada ocupa 1/4 da área ocupada por um caractere. Obviamente, perdemos a possibilidade de "semitons", pois a unidade gráfica do **PLOT** é única. Em compensação os desenhos podem ser muito mais detalhados, pois, a matriz gráfica tem agora

44 X 64

posições.

Este é o máximo de resolução que podemos obter na tela da TV. Em breve, teremos no Brasil, a produção de um periférico para o TK, que permitirá a elaboração de desenhos com alta resolução na tela de TV. Por enquanto devemos nos contentar em fazer estes desenhos na impressora TK-PRINTER.

O programa que apresentamos a seguir, tem uma subrotina que altera o comando **LPRINT** para a impressora, permitindo a impressão de uma matriz de

256 X 256

posições.

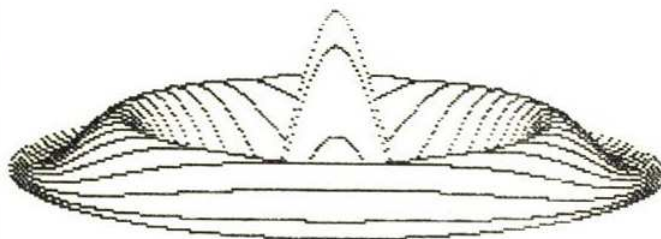
No TK-85 ele roda tranquilo, mas no TK-82 é indispensável uma expansão de pelo menos, 16K.

PROGRAMA DO MÊS
ALTA RESOLUÇÃO NA IMPRESSORA DO TK

Continuação...

```
   1 REM U■RND.U■RND■▘ .TAN ..
   2 POKE 16389,124
   3 CLS
   4 IF NOT PEEK 16388+256*PEEK
16389=31744 THEN STOP
   5 FOR I=0 TO 112
   6 POKE 31744+I,PEEK (2161+I)
   7 NEXT I
   8 POKE 31800,63
   9 POKE 31857,201
  10 POKE 16517,95
  11 POKE 16524,79
  12 DIM A$(32,256)
  13 FAST
  14 LPRINT "    /"
  15 LPRINT "GRAFICO EM 3-D"
  20 LET X1=128
  30 LET X2=X1**2
  40 LET Y1=128
  50 LET Y2=32
  60 FOR B=0 TO 127
  70 LET X4=B**2
  80 LET M=-Y1
  90 LET A=SQR (X2-X4)
 100 FOR I=-A TO A STEP 10
 110 LET R=SQR (X4+I*I)/X1
 120 LET F=COS (9*R)*(1-R)*2
 130 LET Y=I/5+F*Y2
 140 IF Y<=M THEN GOTO 180
 150 LET M=Y
 160 LET Y=Y1+Y
 165 LET X=X1+B
 170 GOSUB 9980
 174 LET X=X1-B
 175 GOSUB 9980
 180 NEXT I
 190 NEXT B
 195 GOTO 9988
9980 REM PLOTA (X,Y) NA MATRIZ
         A$
9981 IF X<0 OR X>255 OR Y<0 OR Y
>255 THEN RETURN
9982 LET C=1+INT (X/8)
9983 LET R=256-INT Y
9984 POKE 16526,CODE A$(C,R)
9985 POKE 16527,2**(8*C-INT X-1)
9986 LET A$(C,R)=CHR$ (USR 16514
)
9987 RETURN
9988 REM  IMPRIME 8 LINHAS DE
         CADA VEZ
9989 FOR I=0 TO 246 STEP 8
9990 FOR J=1 TO 32
9991 FOR K=1 TO 8
9992 POKE 32255+K+8*(J-1),CODE A
$(J,K+I)
9993 NEXT K
9994 NEXT J
9995 FOR H=0 TO 31
9996 POKE 16444+H,H
9997 NEXT H
9998 LET HPRINT=USR 31744
9999 NEXT I
```

EXEMPLO:

GRÁFICO EM 3-D

A linha 1 contém, após o **REM**, uma sequência de caracteres cujos códigos constituem uma sub-rotina em linguagem de máquina.

Esta linha deve ser digitada com todo o cuidado, pois por qualquer errinho põe tudo a perder.

Para facilitar, colocamos a seguir, a lista dos caracteres que a constituem com o número de seu código (em caso de dúvida consulte o apêndice A do manual do TK).

Caractere	Código
U	CODIGO=58
■	CODIGO=142
RND	CODIGO=64
.	CODIGO=27
U	CODIGO=58
■	CODIGO=143
RND	CODIGO=64
■	CODIGO=179
▘	CODIGO=6
	CODIGO=0
.	CODIGO=27
TAN	CODIGO=201
.	CODIGO=27
.	CODIGO=27

Esta sub-rotina permite a superposição de um gráfico sobre outro, fazendo a operação lógica OR entre os dois bytes nos endereços 16526 e 16527. As linhas 2 e 3 reservam um espaço na memória e a linha 4 bloqueia o programa, no caso desta reserva não ter se efetivado. Neste caso é conveniente se salvar o que foi digitado em fita e a seguir dar ao computador os dois comandos.

POKE 16389, 124 (**NEW LINE**)
NEW (**NEW LINE**)

Em seguida, recarregar o programa já digitado no computador.

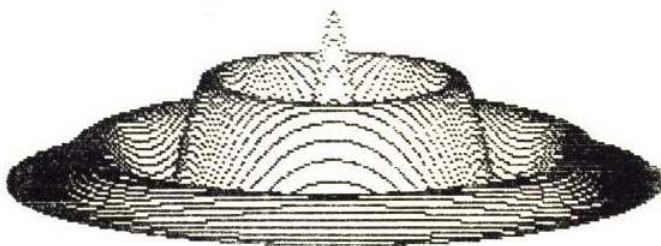
PROGRAMA DO MÊS
ALTA RESOLUÇÃO NA IMPRESSORA DO TK

Continuação...

As linhas seguintes, até a 11 completam a primeira sub-rotina e tomam a sub-rotina **LPRINT** da **ROM** do TK e a alteram para as finalidades deste programa.

A matriz A$, dimensionada na linha 12, ocupa 8K da memória e armazena a posição dos pontos a serem "plotados" na impressora.

A sub-rotina de 998Ø a 9987 registra as coordenadas X e Y na matriz A$, e a subrotina de 9988 a 9999 imprime os pontos (8 linhas de cada vez).

No programa que estamos publicando as linhas de 13 a 195, geram o gráfico tridimensional, que o pessoal de gráfica batizou de "cogumelo". Alterando apenas estas linhas, mas deixando intactas as linhas de 1 a 12 e 998Ø a gráficos em alta resolução. Podem até ser geradas mensagens em maiúsculas e minúsculas, com acentos e cedilhas. No gráfico do cogumelo você poderá alterar o formato mudando as linhas 1ØØ e 12Ø.

```
ESTE GRAFICO FOI OBTIDO ALTERAN-
DO AS LINHAS 100 E 120 PARA:
100 FOR I=A TO -A STEP 2
120 LET F=COS (16*R)*(1-R)*2

CUIDADO,SEU PROCESSAMENTO DEMORA
APROXIMADAMENTE UMA HORA
```

Após rodar o programa, a sequência da linha **REM** fica acrescida automaticamente de outros caracteres.

Aconselhamos, antes de rodar o programa, salvá-lo em fita uma ou duas vezes. Uma vez rodado, sua gravação ficará muito mais demorada, pois ele armazenará também os 8K da matriz A$. Isto é muito bom, quando você não quiser perder muito tempo para gerar a matriz outra vez.

Este processo é demorado, pois você, estará produzindo uma matriz de

256 X 256 = 65536 posições

Obviamente, por mais bonitos que sejam os gráficos obtidos por este programa, você não vai querer ficar o tempo todo desenhando "cogumelos"; para gerar gráficos de sua invenção, é necessário compreender a "arquitetura" da matriz A$.

Ela armazena 256 linhas e 256 colunas, mas na linha 12, seu dimensionamento é

12 **DIM** A$ (32, 256)

ou seja, 256 linhas, mas, apenas 32 colunas.

Isto é fácil de ser explicado: cada coluna de matriz A$ contém 1 byte, ou seja, 8 bits. Como cada ponto está associado a 1 bit temos:

32 bytes X 8 bits/bytes = 256 bits

A posição dos pontos na matriz é dada pelas suas coordenadas (X, Y):

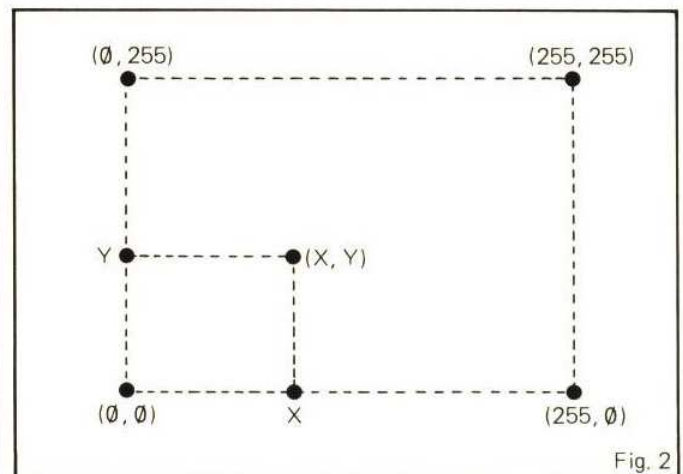



Fig. 2

A posição dos elementos da matriz A$ é dada pela figura 3.

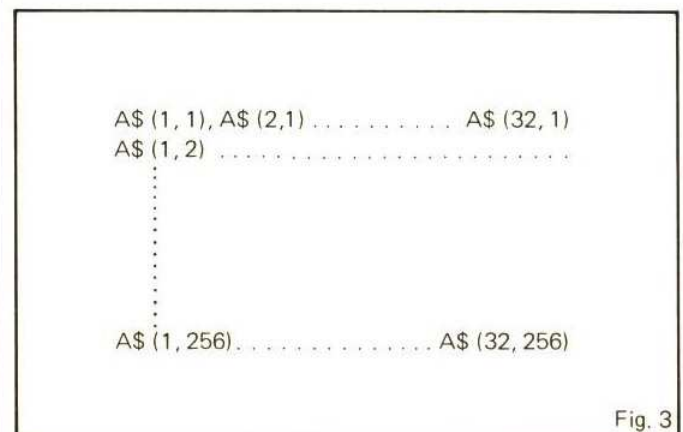



Fig. 3

PROGRAMA DO MÊS
ALTA RESOLUÇÃO NA IMPRESSORA DO TK

Continuação...

Como já dissemos, cada elemento A$ representa 1 linha e 8 colunas (uma por bit).

Vamos recordar um pouco a estrutura de um byte (figura 4).

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit Ø
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Se quisermos, por exemplo, imprimir um elemento qualquer da matriz, pontos nas colunas dos bits 6, 4, 2 e 1, deixando os outros em branco, deveremos atribuir ao byte correspondente o valor

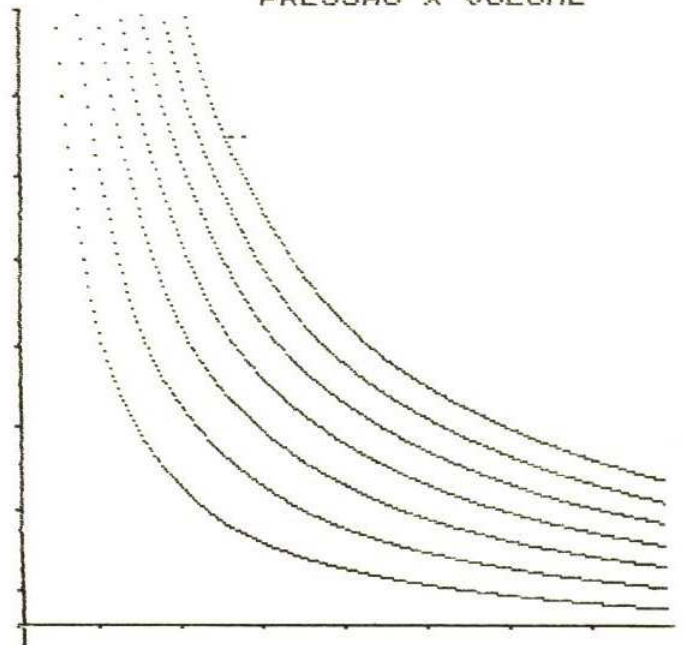
$$64 + 16 + 4 + 2 = 86$$

Assim, se por exemplo, quisermos desenhar no canto superior esquerdo o que está representado na figura 5.

deveremos fazer o seguinte "miolo" de programa:

```
200 LET A$ (1, 1) = CHR$ (169)
210 LET A$ (1, 2) = CHR$ (168)
220 LET A$ (1, 3) = CHR$ (251)
230 LET A$ (1, 4) = CHR$ (33)
240 LET A$ (1, 5) = CHR$ (33)
250 GOTO 9988
```

Deixamos agora à imaginação do leitor, o desenvolvimento de ulteriores aplicações. Apenas a título de exemplo, vamos dar o "miolo" de programa que gera uma série de curvas isotérmicas de um gás no diagrama de Clapeyron.

```
  12 DIM A$(32,256)
  13 FAST
  14 LPRINT "TRANSFORMACOES ISOT
ERMICAS"
  15 LPRINT TAB 12;"PRESSAO X VO
LUME"
  20 FOR I=1 TO 256
  30 LET A$(1,I)=CHR$ (32)
  40 NEXT I
  50 FOR I=1 TO 256 STEP 32
  60 LET A$(1,I)=CHR$ (224)
  62 NEXT I
  64 FOR I=1 TO 32
  70 LET A$(I,239)=CHR$ (255)
  80 NEXT I
  90 FOR I=1 TO 32 STEP 4
 100 LET A$(I,240)=CHR$ (126)
 110 NEXT I
 200 LET U=4096
 210 FOR P=U TO 4*U STEP .5*U
 220 FOR X=9 TO 252
 230 LET Y=P/X+8
 240 GOSUB 9980
 250 NEXT X
 260 NEXT P
 270 GOTO 9988
```

DICAS
A FITA BRINDE

A fita brinde que foi enviada para nossos assinantes contém dois programas interessantes: PULGA para 2K e SIMULADOR DE VÔO para 16K.

A fita está gravada dos dois lados para maior segurança. Com relação ao programa SIMULADOR DE VÔO cabem duas observações:

1) Querendo mudar o "plano de vôo" devemos digitar M (mapa) e a seguir B. A nova cidade escolhida aparecerá piscando no mapa. Digitando V aparecerá novamente a cabine. Nas proximidades do pouso digitar V novamente: a cabine desaparecerá e em seu lugar vemos o campo de pouso em perspectiva.
2) No TK85 há um pequeno grilo facilmente solucionável:
a linha

```
839Ø RETURN
```

deve ser substituída por

```
839Ø GOTO 8391
8391 RETURN
```

Para isto basta dar um **BREAK** no programa e listá-lo. Se não fizermos isto, o programa sumirá de **RAM** após o primeiro acidente (o TK85 é muito sugestionável!).

O programa assim alterado poderá agora ser armazenado em **HIGH-SPEED.**

DICAS
COMO MEMORIZAR A TELA

(Pierluigi Piazzi)

A **RAM** do TK é estruturada de uma maneira um pouco diferente da de outros microcomputadores: o "display", ou seja, a tela não é armazenada num ponto fixo da memória, mas movimenta-se para cima e para baixo em função do tamanho do programa. O endereço da primeira posição do arquivo da imagem é uma variável do sistema denominada de **DFILE** e seu valor, está armazenado nos Bytes dos endereços 16396 e 16397.

Para ler o valor de **DFILE** (**D** = Display: tela – **FILE**: arquivo), basta fazer:

```
PRINT PEEK 16396 + 256 * PEEK 16397
```

Se seu computador estiver "vazio", ou seja, sem nenhum programa armazenado, você obterá

16509

que é o endereço do primeiro byte do programa.

Mas se você fizer:

```
1 PRINT PEEK 16396 + 256 * PEEK 16397
```

e a seguir **RUN**

você obterá

1655Ø

pois o programinha correspondente à Linha 1 que você digitou ocupou 41 bytes e portanto deslocou a memória correspondente ao arquivo da tela de 41 bytes (fig. 1).

COMEÇO DA RAM DO TK

Variáveis do sistema (incluindo DFILE)	Programa de 41 bytes	Arquivo da tela
16384	165Ø9	1655Ø

Vejamos agora, como transferir o conteúdo do arquivo da tela para uma única variável.

Em muitos programas formam-se na tela desenhos ou letreiros gastando para isso muito tempo de proces-

DICAS
COMO MEMORIZAR A TELA

Continuação...

samento. Para reproduzir estes desenhos sem gastar novamente todo tempo de processamento, podemos adotar como sub-rotina o programa a seguir:

```
3000 DIM A$(704)
3010 FOR I=0 TO 21
3020 FOR J=1 TO 32
3030 LET A$(J+32*I)=CHR$ PEEK ((
PEEK 16396+256*PEEK 16397)+J+33*
I)
3040 NEXT J
3050 NEXT I
```

A variável A$ terá 7Ø4 elementos pois

22 linhas x 32 colunas = 7Ø4 posições

Na linha 3Ø3Ø o final é realmente

. . J + **33** * I)

e não 32 pois ao fim de cada linha da tela há um byte de **NEW LINE.** Esta sub-rotina deve ser chamada por alguma instrução do próprio programa e **nunca por um comando direto.**

Toda vez que você dá um comando direto ao seu TK, seu primeiro efeito é justamente apagar a tela. Desta forma ao começar, a sub-rotina copiará uma tela vazia.

Você poderá então intercalar no seu programa uma instrução do tipo

1510 **IF INKEY$** = "P" **THEN GOTO** 3ØØØ

desta forma, ao apertar a tecla P seu programa parará e a tela ficará inteirinha copiada na matriz A$. Para checar basta dar o comando direto

PRINT A$

e seu desenho, que demorou tanto para ser elaborado, reaparecerá em uma fração de segundo!

Dispondo de bastante memória, você pode trocar o A$ das linhas 3ØØØ e 3Ø3Ø por B$ e repetir o processo. Cuidado apenas para **nunca** digitar **RUN** ou **CLEAR!**

Oportunamente você poderá fazer um outro pedaço de programa para imprimir A$, B$, C$ e assim sucessivamente.

Outra vantagem deste processo está no fato de poder arquivar todos os desenhos feitos em fita magnética, pois ao dar um "**SAVE**" você estará armazenando as matrizes A$, B$, . . .

PROGRAMA DO LEITOR

"O POETA"

Bernardo Cláudio Stein

Para pessoas que não estão familiarizadas com computação, o microcomputador parece uma máquina inacessível, misteriosa e fria.

O programa "O POETA" que ora apresentamos aos leitores, foi criado para ser utilizado por pessoas sensíveis e poéticas. Gaste um pouco do seu tempo e digite o programa no seu TK85 ou TK82 com expansão.

Este programa, gera lindas poesias, a partir de 17 expressões que você mesmo escolhe e introduz no computador.

Vejamos como . . .

1) Digite a listagem apresentadas na "figura 1".

2) Passe o programa para o gravador

a) Entre "**GOTO** 32ØØ"

b) Coloque o gravador em "REC"

c) Entre "**NEW LINE**"

3) Após a gravação, o programa apresentará o Menu da "Figura 2".

4) Escolha a Opção número 1.

5) O programa convida você a entrar com 17 expressões à sua escolha.

a) escolha uma palavra (ou conjunto)

b) entre com ela no computador

c) entre "**NEW LINE**"

d) continue com as expressões até completar 17.

6) O programa volta ao Menu principal.

7) Se você quiser rever as palavras escolhidas entre com a Opção 2.

8) Você verá, então, por alguns segundos, os 17 termos escolhidos e depois retorna ao Menu.

9) AGORA O GRANDE MOMENTO. . .

Escolha a Opção número 3.

O computador gerará uma poesia usando as 17 expressões que sua veia poética escolheu.

10) Se você possuir a TK-PRINTER poderá guardar para sempre aqueles melhores poemas que o computador gerou: basta digitar "**COPY**" e "**NEW LINE**" quando a poesia estiver apresentada na tela.

11) Para voltar ao menu principal entre "**GOTO** 321Ø" e "**NEW LINE**".

12) Se você quiser gerar logo outro poema entre "**GOTO** 2" e "**NEW LINE**".

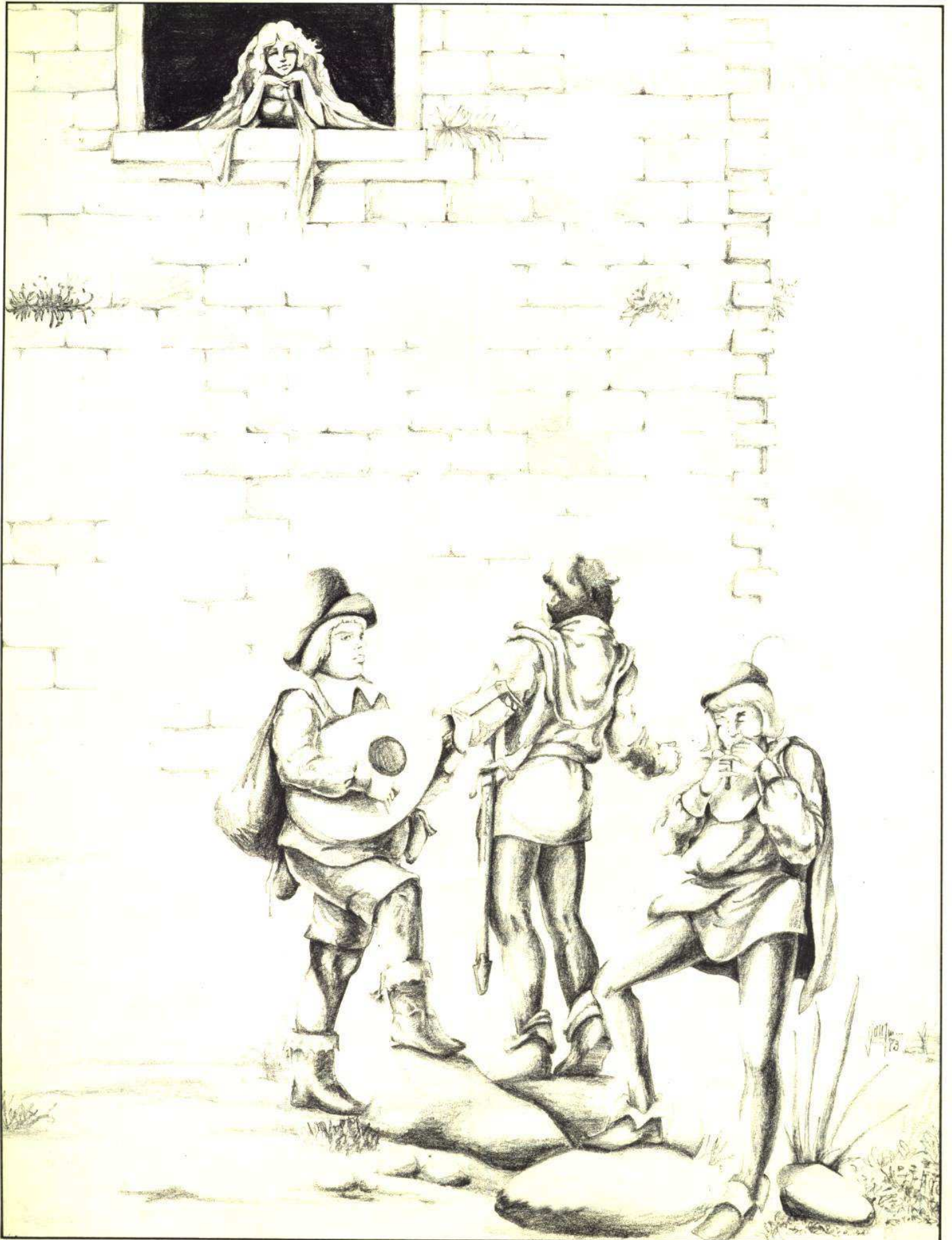
13) Você pode armazenar para sempre o programa **COM** os 17 termos escolhidos:

a) Escolha a "Opção nº 4" do Menu

```
VOCE, COM SUA VEIA POETICA, IRA
CRIAR 17 PALAVRAS QUE SERAO
USADAS POR MIM PARA GERAR LINDOS
POEMAS

ENTRE COM AS PALAVRAS, UMA A UMA

1.

"▦"

PALAVRAS QUE ESTAO NA MEMORIA:

SE APARECER A MENSAGEM DE ERRO
2/3030, ENTRE GOTO 3210, APOS 1.

1.    O SOL
2.    A BRISA
3.    AS ONDAS
4.    BRILHANDO
5.    A CHUVA
6.    CAINDO
7.    AS
8.    HORAS PASSANDO
9.    CREPUSCULO
10.   SEMPRE
11.   NUNCA
12.   COM
13.   SEM
14.   ELE
15.   ELA
16.   BEIJOS
17.   LAGRIMAS

QUANDO A POESIA TIVER PREENCHIDO
A TELA, ENTRE "GOTO 3210"

AS


ELA O SOL
CAINDO BRILHANDO BRILHANDO AS


ELE ELA AS ONDAS
BRILHANDO NUNCA COM HORAS PASSAN
DO LAGRIMAS ELE SEMPRE


AS ONDAS
BEIJOS ...A CHUVA BEIJOS SEMPRE
LAGRIMAS


CAINDO
```

figura 1

b) O programa será então gravado com as palavras escolhidas.

14) Se você quiser carregar o programa na memória do TK entre com **LOAD** "O POETA" ou **LOAD** **" "**

PROGRAMA DO LEITOR "O POETA"

Continuação...

```
              O POETA

   VOCE QUER...

 1.     CRIAR UMA NOVA POESIA

 2.     REVER AS PALAVRAS QUE
          ESTAO NA MEMORIA

 3.     GERAR UMA POESIA COM
            AS PALAVRAS
        QUE ESTAO NA MEMORIA

 4.     GRAVAR AS PALAVRAS
           DA MEMORIA

                             FIG.2
```

15) Durante a utilização do programa evite entrar com "**RUN**", pois isto apagará todos os 17 termos da memória. Neste caso, se você escolher a "Opção nº 2" será apresentada a mensagem de erro 2/3030.

Neste caso entre com "**GOTO** 3210" e "**NEW LINE**" e escolha a "Opção nº 1" do Menu.

Após algumas tentativas, você perceberá a conveniência de utilizar substantivos, já com seus adjetivos, verbos no infinitivo e truques similares. Desta forma você evitará erros de concordância muito evidentes. Certa vez, o físico James Jeans disse que, se pudéssemos acorrentar dois chimpanzés a um piano e eles pudessem bater nas teclas ao acaso indefinidamente, com certeza, após alguns milhões de anos, tocariam a 5ª Sinfonia de Beethoven.

Ajudando um pouco o seu TK, escolhendo locuções bem fortes poeticamente, cercados de expressiva aura conotativa, você poderá gerar poesias incríveis (no meio, obviamente, de ridículos fracassos).

Muito bem, agora mãos à obra.

Quem sabe, você poderá se tornar um novo Olavo Bilac; com a ajuda do seu "TK", obviamente.

```
   1 REM       O POETA
          (C) -B.C.STEIN
             MAR 1983
   2 CLS
   5 PRINT "QUANDO A POESIA TIVE
R PREENCHIDOA TELA. ENTRE ""GOTO
 3210"""
   7 PRINT "▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀
▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
   8 PRINT
  10 LET J=INT (21*RND)
  15 IF J=0 THEN PRINT "...";
  20 IF J>=1 AND J<=3 THEN GOSUB
 1000
  30 IF J>=4 THEN GOSUB 10+J*10
  40 GOTO 10
  50 PRINT A$;" "
  55 RETURN
  60 PRINT B$;" "
  65 RETURN
  70 PRINT C$;" "
  75 RETURN
  80 PRINT D$;" ";
  85 RETURN
  90 PRINT E$;" ";
  95 RETURN
 100 PRINT F$;" ";
 105 RETURN
 110 PRINT G$;" ";
 115 RETURN
 120 PRINT H$;" ";
 125 RETURN
 130 PRINT I$;" ";
 135 RETURN
 140 PRINT J$;" ";
 145 RETURN
 150 PRINT K$;" ";
 155 RETURN
 160 PRINT L$;" ";
 165 RETURN
 170 PRINT M$;" ";
 175 RETURN
 180 PRINT N$;" ";
 185 RETURN
 190 PRINT O$;" ";
 195 RETURN
 200 PRINT P$;" ";
 205 RETURN
 210 PRINT Q$;" ";
 215 RETURN
1000 PRINT
1001 PRINT
1002 PRINT
1003 RETURN
2000 CLS
2005 PRINT AT 6,0;"VOCE, COM SUA
 VEIA POETICA, IRA CRIAR 17 PALA
VRAS QUE SERAO      USADAS POR MI
M PARA GERAR LINDOSPOEMAS"
2010 PAUSE 600
2015 CLS
2020 PRINT "ENTRE COM AS PALAVRA
S, UMA A UMA"
2030 PRINT
2050 PRINT "1.      ";
2053 INPUT A$
2057 PRINT A$
2060 PRINT "2.      ";
2063 INPUT B$
2067 PRINT B$
2070 PRINT "3.      ";
2073 INPUT C$
2077 PRINT C$
2080 PRINT "4.      ";
2083 INPUT D$
```

PROGRAMA DO LEITOR
"O POETA"

Continuação...

```
2087 PRINT D$
2090 PRINT "5.          ";
2093 INPUT E$
2097 PRINT E$
2100 PRINT "6.          ";
2103 INPUT F$
2107 PRINT F$
2110 PRINT "7.          ";
2113 INPUT G$
2117 PRINT G$
2120 PRINT "8.          ";
2123 INPUT H$
2127 PRINT H$

2130 PRINT "9.          ";
2133 INPUT I$
2137 PRINT I$
2140 PRINT "10.         ";
2143 INPUT J$
2147 PRINT J$
2150 PRINT "11.         ";
2153 INPUT K$
2157 PRINT K$
2160 PRINT "12.         ";
2163 INPUT L$
2167 PRINT L$
2170 PRINT "13.         ";
2173 INPUT M$
2177 PRINT M$
2180 PRINT "14.         ";
2183 INPUT N$
2187 PRINT N$
2190 PRINT "15.         ";
2193 INPUT O$
2197 PRINT O$
2200 PRINT "16.         ";

2203 INPUT P$
2207 PRINT P$
2210 PRINT "17.         ";
2213 INPUT Q$
2217 PRINT Q$
2220 PAUSE 270
2230 CLS
2240 GOTO 3210
3000 CLS
3005 PRINT "PALAVRAS QUE ESTAO N
A MEMORIA:"
3010 PRINT "▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀
▀▀▀▀▀▀▀▀"
3020 PRINT "SE APARECER A MENSAG
EM DE ERRO  2/3030, ENTRE GOTO 3
210, APOS 1."
3025 PRINT
3030 PRINT "1.          ";A$
3040 PRINT "2.          ";B$
3050 PRINT "3.          ";C$
3060 PRINT "4.          ";D$
3070 PRINT "5.          ";E$

3080 PRINT "6.          ";F$
3090 PRINT "7.          ";G$
3100 PRINT "8.          ";H$
3110 PRINT "9.          ";I$
3120 PRINT "10.         ";J$
3130 PRINT "11.         ";K$
3140 PRINT "12.         ";L$
3150 PRINT "13.         ";M$
3160 PRINT "14.         ";N$
3170 PRINT "15.         ";O$
3180 PRINT "16.         ";P$
3190 PRINT "17.         ";Q$
3191 PAUSE 540
3192 CLS
3193 GOTO 3210
3194 CLS
3196 PRINT AT 7,2;"COLOQUE O GRA
VADOR EM ""REC""          ENTRE "
"NEW LINE"""
3197 INPUT Z$
3198 CLS
3200 SAVE "O POET▒"

3210 PRINT AT 3,11;"O POETA"
3215 PRINT AT 4,11;"▀▀▀▀▀▀▀"
3220 PRINT AT 7,3;"VOCE QUER..."
3230 PRINT AT 9,1;"1.    CRIAR UM
A NOVA POESIA"
3240 PRINT AT 11,1;"2.    REVER A
S PALAVRAS QUE             ESTAO
 NA MEMORIA"
3250 PRINT AT 14,1;"3.    GERAR U
MA POESIA COM               AS
PALAVRAS                QUE EST
AO NA MEMORIA"
3260 PRINT AT 18,1;"4.    GRAVAR
AS PALAVRAS               DA M
EMORIA"
3270 IF INKEY$="1" THEN RUN 2000
3280 IF INKEY$="2" THEN GOTO 300
0
3290 IF INKEY$="3" THEN GOTO 1
3300 IF INKEY$="4" THEN GOTO 319
4
3310 GOTO 3270
```

DICAS

USANDO AS LINHAS 22 E 23

Flavio Rossini

Quantas vezes você já não deve ter se perguntado se seria possível utilizar as últimas duas linhas da tela (reservadas para a edição), especialmente para a realização de jogos. . .

De fato, isto pode ser feito de um modo bastante simples: na memória de endereço 16418, está uma variável chamada DF-SZ usada pelo TK para indicar o número de linhas reservadas para a região de edição; assim sendo, seu valor normalmente é dois. Basta então modificá-lo para Ø:

1 **POKE** 16418, Ø

Pronto! Experimente fazer agora

2 **PRINT AT** 22, Ø; "BEATLES"
3 **PRINT AT** 23, Ø; "FOREVER"

e **RUN**

No entanto, este truque só pode ser utilizado para **efeitos de tela**! Se você for fazer algum **INPUT** após fazer este **POKE**, "coisas estranhas" poderão ocorrer. Assim, você poderá corrigir DF-SZ antes de fazer um **INPUT**:

POKE 16418, 2

Para exemplificar o efeito, vamos apresentar um programa que ilustra, além disso, alguns truques de programação. Tente explicar como funcionam as linhas 25, 3Ø e 6Ø. . .

```
   1 RAND
   5 SLOW
  10 POKE 16418,0
  15 LET X=0
  20 LET Y=0
  25 LET X=X+2*(INKEY$="8")-2*(I
NKEY$="5")
  30 LET Y=Y+(INKEY$="6")-(INKEY
$="7")
  35 IF X<0 THEN LET X=30
  40 IF X>31 THEN LET X=0
  45 IF Y<0 THEN LET Y=23
  50 IF Y>23 THEN LET Y=0
  55 LET Y$=STR$ Y
  60 LET A=INT (2*RND)*128
  65 IF LEN Y$=1 THEN PRINT AT Y
,X;CHR$ (CODE Y$+A)
  70 IF LEN Y$=2 THEN PRINT AT Y
,X;CHR$ (CODE Y$+A);CHR$ (CODE Y
$(2)+A)
  75 GOTO 25
```

Digite **RUN**... aparece o número Ø no canto superior esquerdo: tente movimentá-lo usando as teclas 5, 6, 7 e 8. O que acontece ao chegar ao fim da tela?

Tente agora este programa:

```
  1 SLOW
  2 POKE 16418,0
  5 IF INKEY$="" THEN GOTO 5
 10 PRINT INKEY$;
 15 GOTO 5
```

Digite **RUN** e a seguir qualquer tecla. . . o que acontece ao digitar **NEW LINE**? ●

PROGRAMA DO LEITOR
CALENDÁRIO

No número Ø da MICROHOBBY, publicamos um pequeno artigo a respeito de calendário, no qual era fornecido um programa.

Este programa permite calcular o dia e o mês no qual cai a PÁSCOA desde que seja conhecido o ano (REVISTA DOS USUÁRIOS DO TK — pág. 6). Pedimos também que os leitores nos enviassem um aperfeiçoamento: "dado o ano, em que dia cai o CARNAVAL?"

Recebemos muitas respostas interessantes para cujos autores enviamos os brindes prometidos. Uma delas foi selecionada para publicação: a de autoria de EDSON SAUER GUIMARÃES.

Com relação à pergunta: "O que aconteceu em Roma no dia 10 de outubro de 1582?" A resposta é "Nada, absolutamente nada!". De fato para ajustar o calendário de Juliano para Gregoriano, no dia 5 de outubro de 1582 sucedeu o dia 15 de outubro de 1582. O dia 10, então, não existiu!

Para os leitores que queiram se aprofundar no tema, aconselhamos a leitura do BOLETIM DA SOCIEDADE ASTRONÔMICA BRASILEIRA — ANO 5, Nº 3 — OUT/DEZ 1982, página 5, onde começa um excelente artigo de autoria de ROBERTO BOCZKO.

●

```
   1 REM EDISON SAUER GUIMARAES
  10 CLS
  20 PRINT "DIGITE OS QUATRO ALG
ARISMOS DO  ANO CUJA PASCOA E CA
RNAVAL VOCE QUER "
  30 PRINT
  40 PRINT "A SEGUIR DIGITE NEW
LINE"
  50 INPUT A
  55 IF A<=1582 THEN GOTO 80
  60 GOTO 90
  80 PRINT
  81 PRINT "SINTO MUITO MAS SO V
ALE O    CALENDARIO GREGORIAN
O"
  85 GOTO 50
  90 LET W=A-INT (A/19)*19+1
 101 CLS
 102 PRINT "EM ";A;" A PASCOA CA
I"
 103 PRINT "NO DOMINGO ";
 105 LET C=INT (A/100)+1
 110 LET X=INT (3*C/4)-12
 115 LET Z=INT ((8*C+5)/25)-5
 120 LET D=INT (5*A/4)-X-10
 125 LET E=11*W+20+Z-X
 190 IF E<0 THEN LET E=E+30
 200 LET E=E-INT (E/30)*30
 300 IF E=24 THEN GOTO 400
 310 IF (E=25) AND (W>11) THEN G
OTO 400
 320 GOTO 410
 400 LET E=E+1
 410 LET N=44-E
 420 IF N<21 THEN LET N=N+30
 430 LET B=D+N
 450 LET B=B-INT (B/7)*7
 500 LET N=N+7-B
 505 PRINT
 510 IF N>31 THEN GOTO 600
 520 PRINT N;" DE MARCO"
 530 GOTO 610
 600 PRINT N-31;" DE ABRIL"
 610 LET M=N+9
 620 LET B=0
 630 IF (A-INT (A/4)*4)<>0 THEN
GOTO 680
 640 IF (A-INT (A/100)*100)=0 TH
EN GOTO 660
 660 LET B=1
 670 LET M=M+1
 680 PRINT
 690 PRINT "OS DIAS DE CARNAVAL
CAEM :"
 700 GOSUB 800
 710 PRINT " DOMINGO"
 720 GOSUB 800
 730 PRINT " SEGUNDA-FEIRA"
 740 GOSUB 800
 750 PRINT " TERCA-FEIRA"
 760 PRINT
 770 PRINT "DIGITE C PARA CONTIN
UAR"
 775 INPUT S$
 780 IF S$="C" THEN GOTO 10
 790 GOTO 770
 800 LET M=M+1
 810 LET S=59
 820 IF B=1 THEN LET S=60
 830 IF M<=S THEN GOTO 860
 840 PRINT M-S;" DE MARCO";
 850 RETURN
 860 PRINT M-31;" DE FEVEREIRO";
 870 RETURN
```

NOVIDADES

O HIGH-SPEED DO TK85

O lançamento do TK85 despertou um grande entusiasmo por uma característica que, irônicamente, não é a mais importante: o teclado.

O primeiro computador a usar teclado de membrana foi o ZX80 (e posteriormente ZX81) da SINCLAIR que encontrou este caminho para baratear mais ainda seu custo de produção e tornar o equipamento acessível a uma faixa de mercado que antes estava excluída do maravilhoso mundo da computação.

No Brasil tivemos o NEZ8000 com teclado de membrana exigindo um bocado de pressão e o TK82 com teclado tipo "toque", consideravelmente mais comodo.

Apesar disto, muitos usuários se queixavam do teclado, pois ao "apertar" a tecla queriam sentir um movimento, para ter um retorno. Para atender a este tipo de solicitação a PROLÓGICA lançou o CP200, versão com SLOW do NEZ8000 expandido, com a diferença de ter um teclado de teclas móveis e um "bip" que indica se elas foram realmente acionadas.

A MICRODIGITAL também atendeu a este tipo de solicitação lançando o TK85, com teclas móveis e um "design" muito atraente.

O preço do TK85, porém, em sua versão mais barata (16K), é quase o dobro do TK82 sem expansão. Será que 14K a mais, um teclado mais cômodo e um aspecto mais bonito justificam este acréscimo? A resposta seria obviamente "não" se o TK85 não tivesse um acréscimo maravilhoso: as funções especiais de armazenamento.

Futuramente pretendemos discutir todas, mas neste número queremos destacar o HIGH-SPEED.

O TK82 e o CP200 enviam e recebem dados da fita cassete com uma velocidade inferior a 600 bits por segundo. O TK85 também opera nesta velocidade, podendo receber programas originalmente concebidos para o TK82, utilizando fitas e gravadores de qualidade média:

Acoplado a um gravador de melhor qualidade, porém, ele pode operar com uma velocidade de transferência de 42ØØ bits por segundo!

Não existe, no mundo, um microcomputador deste porte que transfira dados nesta velocidade. Esta afirmação pode parecer ufanismo, mas no fundo reflete um pouco o orgulho que os brasileiros devem sentir numa ocasião destas: produzimos um computador com uma característica que não tem **similar estrangeiro.**

O número 4200 bits por segundo não consegue transmitir o susto do usuário na hora que ele vê, por exemplo, um SICOM ser transferido para a RAM do TK85 **em alguns segundos.**

A maneira de se conseguir uma fita gravada de maneira especial para o TK85 é muito simples: usando um gravador de boa qualidade e uma fita "low noise", deve-se proceder como segue:

Inicialmente transferir o programa de uma fita normal (300 bits por segundo) para a RAM do TK85 usando o comando **LOAD " "**

Com o cursor em **K** (se necessário deve-se dar um **BREAK** ou um **STOP** no programa) digitar

RAND USR 84Ø5

sem ainda apertar **NEW LINE**.

Colocar a fita virgem no gravador, acioná-lo no modo "REC" (gravar) e deixar a fita correr um pouco.

É conveniente, inclusive, se gravar alguma mensagem de referência a viva voz, tipo "programa SICOM gravado em HIGH-SPEED no dia 20 de junho de 83, três, dois, um, já" colocar o plug no MIC e apertar **NEW LINE**.

O comando digitado chama uma subrotina (batizada HISAVE) em linguagem de máquina que faz o TK85 enviar o programa para a fita na velocidade espantosa de que já falamos.

O padrão das faixas que aparece na tela da TV é completamente diferente do **SAVE** "normal" e não existe o hiato usual de 5 segundos.

Ao terminar a gravação, num tempo surpreendentemente curto, aparece o código de reportagem:

Ø/Ø

no canto inferior da tela.

A fita está corretamente gravada? Verificar isto com um "**LOAD**" é muitas vezes frustrante: a gravação **não estava correta** e o resultado é um computador com a **RAM** vazia. Neste momento o TK85 começa a mostrar outra maravilha: a sub-rotina **HIVERIFY**. Você volta a fita até o início da gravação, digita

RAND USR 8539

e aperta o **NEW LINE**.

Ao apertar a tecla PLAY do gravador a tela mostrará um fundo escuro com alguns riscos brancos horizontais. Ao entrar o programa o padrão de riscos se torna parecido com o **HISAVE**.

➡

NOVIDADES
O HIGH-SPEED DO TK 85

Continuação...

Assim que o programa "passa" pelo computador, sem ser carregado e sem afetar o que ainda está na RAM, a tela pode mostrar três coisas: ou um OK tranquilizador na primeira linha, ou uma mensagem indicando

ERRO

ou continua escura. Neste último caso convém repetir o processo aumentando o volume de reprodução. Se a mensagem indica ERRO é conveniente repetir o processo de **HISAVE**.

Se a mensagem diz "OK", então você pode desligar seu TK tranquilo pois o programa está bem gravado.

Quando quiser recolocar o programa assim gravado na RAM, você deve usar a sub-rotina **HILOAD.**

Para tanto coloca-se a fita no início da gravação, digita-se

RAND USR 863Ø

e **NEW LINE**. Põe-se a fita para tocar no PLAY e espera-se o código Ø/Ø no canto inferior da tela, indicando que o programa já passou da fita para a RAM do TK85.

Resumindo:

HISAVE ➔**RAND USR** 84Ø5:	Usar bom gravador e fita de qualidade. Uma mensagem a viva voz ajuda localizações posteriores.
HIVERIFY➔**RAND USR** 8539:	Compara o programa da fita com o que ainda está na RAM indicando se a gravação foi correta.
HILOAD ➔**RAND USR** 863Ø:	Passa o programa da fita para a RAM, apagando eventuais programas pré-existentes.

Quem já tem uma coleção de fitas gravadas para o TK82, CP200, ZX81, pode agora, com estas três sub-rotinas, preparar outra coleção para o TK85.

Realmente, quem comprou o TK85 para ter um teclado "mais cômodo" vai ter muitas agradáveis surpresas: nem sempre se compra gato por lebre. Este é um típico caso em que muita gente acabou comprando lebre por gato.

E isso porque ainda não falamos de outras maravilhas do TK85 pois queremos deixar alguma coisa para um próximo número de MICROHOBBY.

Por isso não estranhe, se souber de algum turista americano que acabou contrabandeando um microcomputador brasileiro para os Estados Unidos.

●

CURSO DE BASIC AULA 2

Flavio Rossini e Pierluigi Piazzi

Até agora, só mandamos o computador executar uma ordem por vez; como fazer para que ele execute várias ordens em sequência? Para isto, é necessário NUMERAR as ordens (também chamadas instruções). Por exemplo, digite:

```
10 LET A=0█
```

e, a seguir, **NEW LINE**. Pelo fato de ter um **número** na sua frente, a ordem **não é executada;** apenas **copiada** na parte de cima da tela da seguinte maneira:

```
10▶LET A=0
```

a seguir, digite:

```
20 PRINT A█
```

e **NEW LINE**, na tela você irá obter

```
10 LET A=0
20▶PRINT A
```

Notamos que apareceu um novo cursor, o cursor ▶

CURSO DE BASIC

AULA 2

Continuação...

que indica para qual das linhas o computador está "olhando". Para deslocá-lo, utilize **SHIFT** juntamente com as teclas 6 (↓) e 7 (↑). Suponha que você tenha "errado" a instrução (ou linha) 1Ø e quisesse fazer com que a variável A fosse igual a 15 e não Ø. Desloque, então, o cursor ▶ para a linha 1Ø (usando **SHIFT** e a tecla 7); para poder corrigí-la, devemos deslocar o cursor usando → (ou ←), mas isto só é possível, se a linha estiver na parte debaixo da tela. Para trazê-la a esta posição, basta apertar **SHIFT** e a tecla I (EDIT) e você obterá a linha copiada na parte inferior da tela:

```
10K LET A=0
```

Desloque o cursor até depois do dígito Ø, apague-o (usando **RUBOUT**), e substitua-o por 15, você obterá:

```
10 LET A=15L
```

e, a seguir, **NEW LINE**; a linha 1Ø no topo da tela será substituída por esta NOVA linha 1Ø. Vamos fazer voltá-la à original, usando agora outro método; sem trazer a linha para baixo, escreva uma nova linha 1Ø na parte inferior da tela:

```
10 LET A=0L
```

e aperte **NEW LINE**.

*

Novamente, a linha 1Ø superior será alterada.
Para exercitar, tente agora substituir a linha 1Ø por:

```
10 LET A=5
```

utilizando qualquer um dos dois métodos.
Feito isto, vamos acrescentar outra instrução:

```
30 PRINT 3*AL
```

Após apertar **NEW LINE**, teremos então o seguinte:

```
10 LET A=5
20 PRINT A
30>PRINT 3*A
```

Observe que, intencionalmente, não estamos utilizando números consecutivos para numerar as instruções. Isto é útil, para podermos colocar novas instruções **entre** duas já existentes, o que, não seria possível se as duas tivessem números consecutivos. Por exemplo, digite o seguinte:

```
25 PRINT "A=";AL
```

e a seguir **NEW LINE**. Você observará que esta instrução "toma" seu devido lugar entre as linhas 2Ø e 3Ø. O que temos agora? Vejamos, 4 instruções "numeradas"; se executássemos uma de cada vez, iríamos obter respectivamente na tela nada na primeira, pois ela apenas atribui o valor 5 na memória à variável A, na segunda, apareceria o número 5 na tela, na terceira, a letra A seguida do caractere = e do valor da variável A (ou seja 5) e na quarta, o número 15 (3 * 5); mas, agora, poderemos executá-las TODAS sequencialmente. Para isto basta pressionar a tecla R; como o cursor está em K, aparecerá a "key-word":

```
RUN L
```

e a seguir **NEW LINE**. Esta instrução tem o seguinte efeito: faz inicialmente um **CLEAR** (para limpar todas as variáveis que eventualmente tinham sido definidas) e, a seguir, ordena ao computador que ele execute sequencialmente as ordens numeradas. Portanto na tela você irá obter:

```
5
A=5
15
```

e na memória teremos o número 5 associado à variável A. Note o efeito do ponto e vírgula (;) na instrução 25. Ele indica que duas coisas serão escritas "lado a lado" com um só **PRINT**.

Preste atenção: quando você numera as ordens, o computador NÃO as esquece. Em outras palavras, as ordens numeradas são armazenadas em sua memória. Neste instante, você não as vê na tela pois ele está apenas mostrando os RESULTADOS das ordens. Para fazer com que ele mostre as ordens na tela, utilize a seguinte instrução:

```
LIST L        (letra K)
```

e **NEW LINE**. O que acabamos de fazer é muito importante e precisa estar bem claro em sua cabeça: ao ordenar ao computador que faça uma determinada tarefa, **sem** colocar um número na frente da ordem, o computador a executa IMEDIATAMENTE esquecendo a ordem em seguida; se você numerar as ordens, elas são executadas ao receber o comando **RUN** e o computador NÃO as esquece. Elas estão em sua memória e podem ser vistas na tela fazendo um **LIST**. A fase em que você escreve as ordens no computador é chamada de PROGRAMAÇÃO; ao apertar a tecla **RUN**, ele executará as ordens do seu PROGRAMA, fase essa, que chamamos de EXECUÇÃO. É importante não confundir as duas coisas.

Vamos agora "brincar" um pouco com o nosso programa; acrescente ao mesmo a instrução:

→

CURSO DE BASIC

AULA 2

Continuação...

```
40 LIST
```

e execute este "novo programa", apertando **RUN** e **NEW LINE**. Veja, o computador irá executar as suas ordens, sendo que a última, diz para que ele MOSTRE o programa na tela, assim você irá obter:

```
5
A=5
15
  10 LET A=5
  20 PRINT A
  25 PRINT "A=";A
  30 PRINT 3*A
  40 LIST
```

Ou seja, no programa você ordenou que ele **se mostrasse** no final. Substitua agora a linha 4Ø por:

```
40 RUN
```

E execute o programa fazendo **RUN** e **NEW LINE**.

O que você obtém? Várias vezes a mesma saída 5, A e 15. Você saberia explicar por que? Observe a sua última instrução: você, ao terminar o programa, ordena que ele comece de novo. Dessa forma, ele irá se repetir até não haver mais espaço na tela. Note que a ordem de **RUN** indica ao computador que ele comece a executar o programa, a partir da primeira linha. Você pode fazer com que ele execute, a partir de qualquer outra linha. Faça **LIST** e **NEW LINE** e, a seguir, substitua a linha 4Ø por:

```
40 RUN 30
```

a seguir, digite **RUN** e **NEW LINE**.

Veja o que acontece: o computador "escreve" na tela 5, A e 15 apenas uma vez, pois **RUN** 3Ø significa "dê um **CLEAR**", e execute o programa a partir da linha 3Ø. Ora, ao tentar executar a instrução 3Ø ele irá procurar a variável A na memória, mas não irá encontrá-la, pois ela foi apagada pelo **CLEAR**; assim, ele pára a execução do programa. Experimente então, substituir a linha 3Ø por:

```
30 LET A=8
```

e acrescente a linha 35 da seguinte maneira:

```
35 PRINT 3*A
```

Execute então o programa assim modificado. Tente explicar o que ocorre.

```
10 LET A=5
20 PRINT A
25 PRINT "A=";A
30 LET A=8
35>PRINT 3*A
40 RUN 30
```

```
5
A=5
24
24
24
24
24
24
24
24
24
24
24
24
24
24
24
24
24
24
24
24
```

Da mesma forma, a instrução **LIST**, mostra o programa na tela a partir da primeira linha; experimente ordenar:

```
LIST 25
```

e **NEW LINE**. Observe que o computador mostra o seu programa na tela a partir da linha 25.

```
25>PRINT "A=";A
30 LET A=8
35 PRINT 3*A
40 RUN 30
```

Acrescente ao seu programa agora, 2 linhas:

```
22 PRINT
27 PRINT
```

de tal maneira que ele fique:

```
10 LET A=5
20 PRINT A
22 PRINT
25 PRINT "A=";A
27 PRINT
30 LET A=8
35 PRINT 3*A
40 RUN 30
```

CURSO DE BASIC

AULA 2

Continuação...

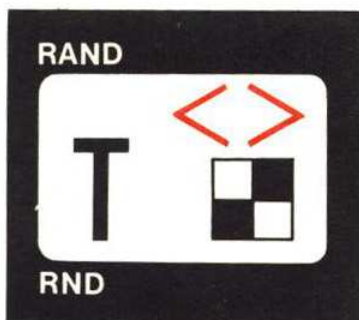
e mande agora executar o seu programa (**RUN** e **NEW LINE**). Você irá obter uma linha em branco, entre o número 5 e a letra A, e, entre a letra A e o primeiro número 24, pois, de fato, você ordenou que ele NÃO escrevesse nada.

Dessa forma, você conseguiu fazer com que ele "pulasse" linhas.

Cabe aqui um comentário com relação à instrução **CLEAR**: ela apenas limpa as VARIÁVEIS do computador. De fato, experimente ordenar **CLEAR** (e **NEW LINE**) e, a seguir faça **LIST**. Você pode observar que seu programa ainda está lá. Se você desejar "limpar" (apagar) uma linha de programa, basta colocar o número da linha, seguido de **NEW LINE**; experimente escrever 4Ø e **NEW LINE**. A linha 4Ø irá desaparecer. Se você quiser apagar TODO O PROGRAMA e TODAS AS VARIÁVEIS, basta executar o seguinte comando:

NEW (tecla A)

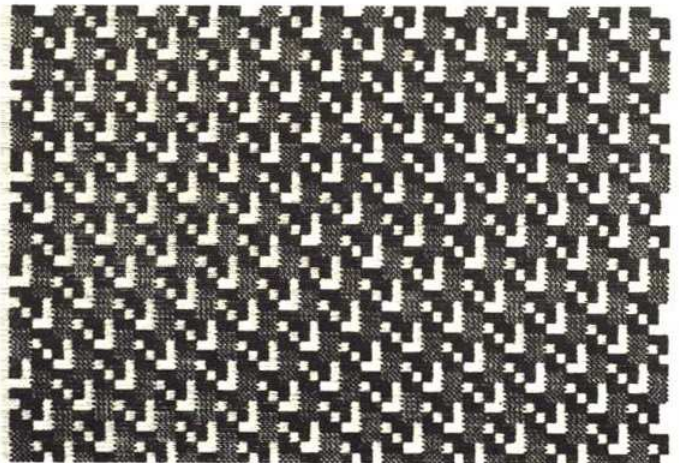
e **NEW LINE**. Experimente "listar" o seu programa; nada aparecerá na tela, ele "esqueceu" tudo. . . Portanto, cuidado com esta instrução: ela é **EXTREMAMENTE PERIGOSA.**

Para finalizar, vamos aprender mais um cursor: trata-se do cursor **G** (GRAPHICS) cujo acesso é feito pressionando **SHIFT** e a tecla 9. Observe, por exemplo, a tecla T:

Se o cursor estiver em **K**, ao pressioná-la, você obtém **RAND** na tela; com o cursor **L**, você terá T, ou <> se pressionar simultaneamente o **SHIFT**; com o cursor em **F**, você obtém **RND**; agora, com o cursor em **G** ao pressioná-la, você terá a letra T em vídeo reverso, ou, o desenho ▚, se pressionar simultaneamente o **SHIFT**. Façamos então o seguinte programa:

```
10 PRINT "▞▀▀▘▌"
20 RUN
```

Note que, após abrir as aspas, devemos colocar o cursor em **G** e, simultaneamente com **SHIFT**, apertar as teclas T, H, 7, F e 8. Para fechar as aspas, devemos passar novamente o cursor para **L**; para isto, basta fazer **SHIFT** e a tecla 9 (GRAPHICS). Execute então o programa. A seguir, modifique a linha 1Ø acrescentando um (;) no final:

```
10 PRINT "▞▀▀▘▌";
20 RUN
```

Execute o programa e perceba o efeito "ESTRANHO" do (;) ou seja: faça com que o próximo **PRINT** escreva "ao lado" e não na linha de baixo. Mais adiante veremos alguns detalhes a respeito do (;).

Exercício:

1) Faça um programa que associe à variável **HOPE** o número 6 (na memória) usando a instrução **LET** e, a seguir, escreva na tela o seguinte:

```
HOPE = 6
HOPE + 2 = 8
3*HOPE = 18
```

→

CURSO DE BASIC

AULA 2

Continuação...

```
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
5+7  =  12
```

até o fim da tela.

2) "Apague" o programa anterior (**NEW**) e, a seguir, faça a seguinte instrução (não numerada): **LET** B = 1Ø (**NEW LINE**). Escreva então este programa:

```
100 LET A=5
200 PRINT A
300 PRINT B
```

Note o que aparece na tela ao tentar executá-lo. Explique o que ocorre e a seguir acrescente a linha 15Ø: 15Ø **CLEAR**; tente então, executar o programa e explicar o que acontece. Substitua então a linha 15Ø por: 15Ø **LET** B = 7 e execute o programa. Finalmente, acrescente a linha 4ØØ: 4ØØ **NEW**.

Execute o programa e tente listá-lo em seguida. Explique o que ocorre. ●

CURSO DE ASSEMBLY LINGUAGEM DE MÁQUINA

AULA 1

Flavio Rossini

Para que possamos entender o que é a linguagem de máquina, são necessários alguns conceitos básicos fundamentais:

a) números binários e hexadecimais
b) bit e byte

Além deles, para utilizar o ASSEMBLY com o TK, é preciso um bom conhecimento do "funcionamento" da key-word **POKE** e da função **PEEK**.

O objetivo desta primeira aula é 'portanto' abordar estes tópicos e apresentar alguns exercícios cuja resposta será publicada no próximo número.

"Números binários e hexadecimais"

A maneira pela qual nós estamos acostumados a representar os números, usa dez símbolos diferentes (a saber: Ø, 1, 2, 3, 4, 5, 6, 7, 8 e 9) chamados algarismos, sendo que cada algarismo tem "peso" tanto maior quanto mais a esquerda ele estiver na representação do número. A relação entre o "peso" de algarismos adjacentes é sempre 1Ø; assim, por exemplo:

posição

$1983 = 1 \times 1Ø^3 + 9 \times 1Ø^2 = 8 \times 1Ø^1 + 3 \times 1Ø^Ø$

algarismos

posição: 3 | 2 | 1 | Ø

peso: $1Ø^3$ | $1Ø^2$ | $1Ø^1$ | $1Ø^Ø$

CURSO DE ASSEMBLY LINGUAGEM DE MÁQUINA

Continuação...

Por estas razões, dizemos que os números estão representados na **base 1Ø.** Diz a história que isso se deve ao fato do ser humano ter 1Ø dedos nas mãos . . .

Óbviamente, qualquer base poderia ter sido escolhida e quaisquer símbolos poderiam ter sido adotados; imagine, por exemplo, um marciano com 8 dedos em cada mão . . . ou um venusiano com um dedo apenas em cada mão . . . Apesar dos computadores não serem feitos em Marte ou Vênus, é conveniente representar os números na base **2** (números binários) e na base **16** (números hexadecimais) quando pensamos em trabalhar em linguagem de máquina, por razões que veremos mais adiante; assim, na base **2** necessitamos 2 símbolos para representar os números e na base **16** são necessários 16 (vide tabela I.1); a relação entre o peso de 2 algarismos adjacentes será respectivamente **2** e **16**.

Tabela I.1 — Símbolos usados para representar números binários (base 2), decimais (base 1Ø) e hexadecimais (base 16), e utilização dos símbolos da base 2 e base 1Ø para representar os símbolos da base 16.

BASE		
2 ou B	10 ou D	16 ou H
Ø	Ø	Ø
1	1	1
1Ø	2	2
11	3	3
1ØØ	4	4
1Ø1	5	5
11Ø	6	6
111	7	7
1ØØØ	8	8
1ØØ1	9	9
1Ø1Ø	1Ø	A
1Ø11	11	B
11ØØ	12	C
11Ø1	13	D
111Ø	14	E
1111	15	F

Note que na base 16 somos obrigados a utilizar **letras** para os últimos 6 símbolos, pois nós seres humanos temos apenas 1Ø símbolos numéricos (Ø, 1, ... , 9). Assim, nos números hexadecimais, as letras A, B, C, D, E e F representam com um **único símbolo** os números 1Ø, 11, 12, 13, 14 e 15 da base 1Ø.

Obs.: para evitar confusão, colocaremos os números entre parênteses com a base utilizada em baixo à direita:

$$(\text{número})_{base}$$

Por exemplo, vamos escrever 2 números na base 2 (ou B) e 2 números na base 16 (ou H) e sua respectiva representação na base 1Ø (ou D):

$(11Ø1)_2 = 1 \times 2^3 + 1 \times 2^2 + Ø \times 2^1 + 1 \times 2^Ø = (13)_{1Ø}$

posição: 3/2/1/Ø

peso: $2^3/2^2/2^1/2^Ø$

$(1ØØØ\ 1Ø1Ø)_B = 1 \times 2^7 + Ø \times 2^6 + Ø \times 2^5 + Ø \times 2^4 + 1 \times 2^3 + Ø \times 2^2 + 1 \times 2^1 + Ø \times 2^Ø = (138)_D$

posição: 7/6/5/4/3/2/1/Ø

peso: $2^7/2^6/2^5/2^4/2^3/2^2/2^1/2^Ø$

$(132)_{16} = 1 \times 16^2 + 3 \times 16^1 + 2 \times 16^Ø = (3Ø6)_{1Ø}$

posição: 2/1/Ø

peso: $16^2/16^1/16^Ø$

$(A7)_H = 1Ø \times 16^1 + 7 \times 16^Ø = (167)_D$
(A)

posição: 1/Ø

peso: $16^1/16^Ø$

Note que no último exemplo o símbolo **A** foi substituído pelo número 1Ø para que pudéssemos fazer a conta de modo "convencional" e obter o número $(A7)_H$ na base 1Ø. O mesmo ocorre para B, C, D, E e F que foram usados na base 16 para poder representar com um **único símbolo** os números de 1Ø a 15 que são representados com **dois símbolos** na base **1Ø**.

Assim, vimos como "passar" números representados na base 2 e na base 16 para a base 1Ø. Para passar da base 2 para a 16 ou vice-versa o procedimento é bem simples pois $16 = 2^4$.

Desse modo, dado um número na base 2, para obter sua representação em hexadecimal, basta agrupar os algarismos em grupos de 4 a partir da direita e substituí-los pelo correspondente algarismo na base 16. Você saberia explicar por que?

CURSO DE ASSEMBLY LINGUAGEM DE MÁQUINA

Continuação...

```
(1Ø1ØØ1ØØ11Ø11ØØ)B = (5 2 6 C)H
 5   2   6   C

(11 11Ø1 Ø111 1111)2 = (3D7F)16
 3   D    7    F
```

Obs.: veja a tabela I.1.

Vice-versa, dado um número na base 16, para obter sua representação em binário, basta substituir cada algarismo pela sua correspondente representação em base 2. Note que são **sempre** necessários 4 algarismos da base 2 para cada símbolo da base 16 sendo que os "espaços" devem ser preenchidos com zeros à esquerda:

```
(3BF)16 = (11 1Ø11 1111)2
           3   B    F

(A482)H = (1Ø1Ø Ø1ØØ 1ØØØ ØØ1Ø)B
            A    4    8    2
```

Para completar o "ciclo" das transformações de bases, resta apenas aprender a passar números da base 1Ø para as bases 2 e 16. Via de regra, o seguinte algoritmo pode ser utilizado para passar números na base 1Ø para qualquer base X: (tente explicar por que).

a) divida o número dado por X, anote o resto e o resultado
b) divida o resultado por X e anote o resto e o novo resultado
c) repita a operação (b) até o resultado obtido ser zero (obviamente se o resultado do passo (a) for zero não são necessários os passos (b) e (c)).
d) copie os "restos" sequencialmente começando pelo último.

Exemplo:
Vamos passar alguns números na base 1Ø para a base 2:

```
(35)1Ø = (?)2
35 |2
 1  17 |2
     1  8 |2
        Ø  4 |2
           Ø  2 |2
              Ø  1 |2
                 1  Ø      (35)1Ø = (1ØØØ11)2
```

```
(92)D = (?)B
92 |2
 Ø  46 |2
     Ø  23 |2
         1  11 |2
             1  5 |2
                1  2 |2
                   Ø  1 |2
                      1  Ø  (92)D = (1Ø1 11ØØ)B
```

e da base 1Ø para a base 16:

```
(88)1Ø = (?)16
88 |16
 8   5 |16
     5  Ø                 (88)1Ø = (58)16

(216)D = (?)H
216 |16
 56  13 |16
  8  13  Ø                (216)D = (D8)H
```

(Lembre-se que $(13)_{1Ø} = (D)_{1Ø}$: veja tabela I.1)

Obs.: Note a conveniência de se agrupar os algarismos dos números binários de 4 em 4 para facilitar a compreensão, assim como fazemos para os números decimais (de 3 em 3).

Exercícios:

1) Efetue as seguintes conversões:
 a) $(1Ø\ 11Ø1)_B = (?)_D$
 b) $(1Ø1Ø)_{1Ø} = (?)_{16}$
 c) $(A2F)_H = (?)_B$
 d) $(38Ø)_{16} = (?)_{1Ø}$
 e) $(1Ø1\ ØØ1Ø\ 1Ø11)_2 = (?)_{16}$
 f) $(111\ Ø11Ø)_2 = (?)_{1Ø}$
 g) $(A1ØF)_H = (?)_D$
 h) $(1Ø\ 111Ø\ Ø11Ø\ 11ØØ)_B = (?)_H$
 i) $(ACF2)_{16} = (?)_2$
 j) $(127)_{1Ø} = (?)_2$
 k) $(999)_D = (?)_H$
 l) $(11Ø)_D = (?)_B$

Sugestão para uma tarde de verão: Algum tempo atrás era bastante usada também a base 8 (números octais); tente imaginar números representados nesta base e como transformá-los para as bases que aprendemos hoje e vice-versa (note que $8 = 2^3$); a seguir faça o mesmo para uma base que não seja potência de 2 (ex. 5).

CURSO DE ASSEMBLY
LINGUAGEM DE MÁQUINA

Continuação...

Sugestão para 5 tardes de verão: Experimente imaginar números na base 12 e fazer contas com eles: soma, subtração, multiplicação, divisão . . . Verifique que certas contas de divisão que dão dízima na base 1Ø não dão dízima na base 12; ex.: 4/3 = 1,333 . . . (base 1Ø).
4/3 = 1,4 (base 12)

"Bit/Byte: circuitos eletrônicos de dois estados"

A evolução da eletrônica tornou possível a elaboração de circuitos extremamente pequenos capazes de assumir 2 estados estáveis, sinalizados pela presença ou não de tensão elétrica; para efeito de representação, estes estados são simbolizados respectivamente por **1** e **Ø**, razão pela qual torna-se necessário representar os números na base 2 se quisermos **interpretar** como **números** os estados dos circuitos. Os computadores são constituídos em sua grande parte por **"milhares"** destes circuitos inteligentemente ligados.

Na linguagem "usual", cada algarismo Ø ou 1 é chamado de "bit" e a um conjunto de 8 bits chamamos "byte". Os números hexadecimais foram introduzidos para facilitar a escrita e compreensão de números binários, substituindo longas cadeias de zeros e uns, pelos correspondentes símbolos hexadecimais, reduzindo assim, 4 vezes seu tamanho e facilitando o entendimento do ponto de vista **humano**.

Assim, se tivermos **8** circuitos iguais ao mencionado anteriormente, poderemos representar 256 números (2^8) correspondendo a todas as combinações possíveis de zeros e uns:

(binários)	(decimais)	(hexadecimais)
"ØØØØ ØØØØ"	Ø	'Ø'
"ØØØØ ØØØ1"	1	'1'
"ØØØØ ØØ1Ø"	2	'2'
"ØØØØ ØØ11"	3	'3'
.	.	.
.	.	.
.	.	.
"1111 11ØØ"	252	'FC'
"1111 11Ø1"	253	'FD'
"1111 111Ø"	254	'FE'
"1111 1111"	255	'FF'

Assim, utilizando 1 byte (8 bits) podemos representar os números inteiros positivos de Ø a 255 . . .

Se usássemos **16** bits (ou seja 2 bytes), poderíamos representar 2^{16} números a saber de Ø a 65535. Esta regra é análoga à base dez; de fato, com **2** algarismos poderemos representar $1Ø^2$ números (Ø a 99).

Note que introduzimos aqui uma nova simbologia para diferenciar as bases nas quais estão representados os números: os binários entre aspas ("), os hexadecimais entre apóstrofos (') e os decimais sem "nada"; esta simbologia é a que será adotada para todas as outras aulas.

Como veremos no decorrer do curso, serão muito interessantes para nós os números de 1 byte (8 bits) e 2 bytes (16 bits). Assim como nos números decimais onde existe o conceito de "algarismo mais significativo".

Por exemplo o número hexadecimal:

```
                  1 byte       1 byte
'1A2C' = "ØØØ1 1Ø1Ø  ØØ1Ø 11ØØ"
           1    A     2    C
```

tem como byte mais significativo '1A' e como byte menos significativo '2C'. Este número poderia ser assim representado em decimal:

$$'1A2C' = 1 \times 16^3 + 1Ø \times 16^2 + 2 \times 16^1 + 12 \times 16^Ø = \mathbf{67ØØ}$$

(A) (C)

Poderíamos porém, calculá-lo de outra forma:

byte mais significativo: $'1A' = 1 \times 16^1 + 1Ø \times 16^Ø = 26$

byte menos significativo: $'2C' = 2 \times 16^1 + 12 \times 16^Ø = 44$

e calcular:

(byte mais significativo) $\times 256^1$ + (byte menos significativo) $\times 256^Ø = 26 \times 256^1 + 44 \times 256^Ø = 67ØØ$

Como se estivéssemos utilizando uma base "fictícia" de 256 (= 16^2). Para entender melhor, suponha por exemplo o número 1Ø24 em decimal, ele equivale a:

$$1 \times 1Ø^3 + Ø \times 1Ø^2 + 2 \times 1Ø^1 + 4 \times 1Ø^Ø$$

Se "dividirmos" o número ao meio [1Ø 24] poderemos usar uma base fictícia de $1Ø^2 = 1ØØ$; de fato:

$$1Ø \times 1ØØ^1 + 24 \times 1ØØ^Ø = 1Ø24.$$

Não se trata de coincidência. É lógica . . .

Medite um pouco sobre estes conceitos. Eles serão úteis e necessários para entender a linguagem de máquina. . .

Para facilitar as coisas, apresentamos aqui uma tabela dos primeiros 256 números hexadecimais e decimais para conversão, ou seja, todos os números possíveis de serem representados com 1 byte e que, portanto, têm dois dígitos hexadecimais.

→

CURSO DE ASSEMBLY
LINGUAGEM DE MÁQUINA

Continuação...

Tabela I.2

	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø	Ø	1	2	3	4	5	6	7	8	9	1Ø	11	12	13	14	15
1	16	17	18	19	2Ø	21	22	23	24	25	26	27	28	29	3Ø	31
2	32	33	34	35	36	37	38	39	4Ø	41	42	43	44	45	46	47
3	48	49	5Ø	51	52	53	54	55	56	57	58	59	6Ø	61	62	63
4	64	65	66	67	68	69	7Ø	71	72	73	74	75	76	77	78	79
5	8Ø	81	82	83	84	85	86	87	88	89	9Ø	91	92	93	94	95
6	96	97	98	99	1ØØ	1Ø1	1Ø2	1Ø3	1Ø4	1Ø5	1Ø6	1Ø7	1Ø8	1Ø9	11Ø	111
7	112	113	114	115	116	117	118	119	12Ø	121	122	123	124	125	126	127
8	128	129	13Ø	131	132	133	134	135	136	137	138	139	14Ø	141	142	143
9	144	145	146	147	148	149	15Ø	151	152	153	154	155	156	157	158	159
A	16Ø	161	162	163	164	165	166	167	168	169	17Ø	171	172	173	174	175
B	176	177	178	179	18Ø	181	182	183	184	185	186	187	188	189	19Ø	191
C	192	193	194	195	196	197	198	199	2ØØ	2Ø1	2Ø2	2Ø3	2Ø4	2Ø5	2Ø6	2Ø7
D	2Ø8	2Ø9	21Ø	211	212	213	214	215	216	217	218	219	22Ø	221	222	223
E	224	225	226	227	228	229	23Ø	231	232	233	234	235	236	237	238	239
F	24Ø	241	242	243	244	245	246	247	248	249	25Ø	251	252	253	254	255

Assim, por exemplo, linha 5, coluna A, teremos:

'5A' = 9Ø (de fato $5 \times 16^1 + 1Ø \times 16^Ø = 9Ø$)

Vice-versa, 247 corresponde a linha F, coluna 7, ou seja:

247 = 'F7'

"O que é POKE? O que é PEEK?"

Como veremos, a memória do TK pode, com expansão de 64K, ter 65536 "casinhas" (posições) numeradas de Ø a 65535, cada uma capaz de armazenar 1 byte, ou seja, um número compreendido entre Ø e 255 ('ØØ' e 'FF'); a instrução **POKE** coloca um número (de Ø a 255) na memória desejada; por exemplo:

POKE 3ØØØØ, 135

coloca o número 135 na memória 3ØØØØ. A função **PEEK** nos "conta" qual o número que está em dada memória; por exemplo:

PRINT PEEK 3ØØØØ

escreve na tela o que está na memória 3ØØØØ. Lembre-se, os endereços só vão de Ø a 65535 e o número máximo que cabe em uma memória é 255. Assim, para colocar na memória um número maior que 255 deveremos "quebrá-lo" em várias partes usando a base 256 (16^2), pois ele não cabe em apenas uma posição. Se ele for menor do que 65536 (256^2) devemos "quebrá-lo" em **dois;** se o número for maior que 65536 e menor do que 16777216 (256^3), deveremos quebrá-lo em 3 e assim por diante.

Suponha, por exemplo, que você queira colocar o número 522 nas memórias 3ØØØØ e 3ØØØ1; ora, 522 = = 'Ø2ØA' { 'Ø2' = 2 ; 'ØA' = 1Ø }

Vamos colocar o byte **menos significativo** na memória de **menor endereço:**

POKE 3ØØØØ,1Ø
POKE 3ØØØ1,2

Para obtê-lo de volta, basta fazer:

PRINT PEEK 3ØØØØ + 256 x **PEEK** 3ØØØ1

Experimente agora colocar na memória o número 32477 e obtê-lo de volta (use os mesmos endereços).

Faça o mesmo para o número 65535. . . E, se você estiver inspirado, tente colocar os números 1ØØ.ØØØ e 2Ø.ØØØ.ØØØ na memória.

QUEBRA-CABEÇA

VERDADEIRO OU FALSO?

A partir deste número, inauguramos a seguinte sistemática para resposta do quebra-cabeça: ele será respondido sempre no segundo número após sua publicação. Isto significa que o quebra-cabeça proposto no número 1 terá sua resposta publicada no número 3. Este que estamos propondo agora no número 2, será respondido no número 4 e assim sucessivamente. Neste número 2 começamos também a contar com a colaboração dos membros da MENSA residentes no BRASIL.

O quebra-cabeça que vamos propor agora é de autoria de Jim Geschwender e foi publicado na seção GAMES da revista americana OMNI no mês de maio de 1983 (resposta na página 148). Neste momento alguns leitores se perguntarão: "Mas, qual é a graça?

Basta achar esta revista e procurar a resposta!" Pois é nesse ponto que eles vão ficar surpresos: o quebra-cabeça deste mês não consiste em achar a resposta. Vamos então ao quebra-cabeça e depois dizemos o que pretendemos:

A lista a seguir contém 1Ø asserções que podem ser verdadeiras (V) ou falsas (F).

Assinale V ou F em todas elas de maneira que suas respostas satisfaçam simultaneamente todos os ítens da lista sem contradições:

1. ( ) A quarta asserção falsa tem numeração ímpar.
2. ( ) Há mais asserções falsas entre as primeiras seis (de 1 a 6) que entre as últimas 7 (de 4 a 10).
3. ( ) Existem pelo menos 5 asserções falsas.
4. ( ) Existem três asserções verdadeiras consecutivas.
5. ( ) Existem pelo menos tantas asserções verdadeiras com numeração par, quantas asserções falsas com numeração ímpar.
6. ( ) As asserções 7 e 10 são, ou ambas falsas, ou ambas verdadeiras.
7. ( ) Das três últimas asserções com numeração par (6, 8 e 10) apenas uma é verdadeira.
8. ( ) As asserções 6 e 9 podem ser verdadeiras somente se a asserção 10 for falsa.
9. ( ) A asserção 7 e a primeira asserção verdadeira com numeração par são precedidas imediatamente por asserções verdadeiras.
10. ( ) A asserção 9 é falsa.

Agora você vai quebrar a cabeça tentando achar a resposta ou, se for preguiçoso, vai procurá-la na revista OMNI.

Mas, o que queremos realmente que você faça é "bolar" um programa em BASIC — TK que faça seu computador achar a resposta.

Lembre-se de que existem no BASIC funções e instruções extremamente poderosas do ponto de vista lógico:

QUEBRA-CABEÇA VERDADEIRO OU FALSO?

Continuação...

IF . . . THEN
OR
AND
NOT

Além disso, seu TK tem a seguinte capacidade: qualquer asserção entre parênteses, se for verdadeira a ela atribuimos o valor 1, se for falsa Ø. Experimente digitar:

PRINT (2 = 2)

e

PRINT (47 > 51)

e veja o que seu TK responde em cada caso.

Agora, mãos à obra e envie-nos o programa (veja seção "Como colaborar com MICROHOBBY") a tempo, para que possamos eventualmente publicá-lo no número 4. Boa sorte e bom divertimento.

LANÇAMENTO

LIVRO: APLICAÇÕES SÉRIAS PARA TK82-C

No momento em que estávamos fechando esta edição, chegou às nossas mãos um exemplar do livro APLICAÇÕES SÉRIAS PARA TK82-C E CP200 de autoria de DELIO SANTOS LIMA.

Não tivemos, portanto, tempo material para fazer uma análise detalhada, mas, numa rápida leitura pudemos perceber que este livro contém programas muito interessantes e algumas "dicas" realmente muito úteis para os usuários dos computadores que usam BASIC-TK.

Faremos futuramente uma análise crítica mais profunda, assim como de todos os livros referentes ao TK que nos forem enviados para resenha, mas por enquanto queremos nos congratular com o DELIO por mais esta iniciativa pioneira e bem brasileira. ●

COMO COLABORAR COM MICROHOBBY

A revista MICROHOBBY foi criada para servir de intercâmbio entre os leitores que participam do mágico mundo da computação.

A característica realmente inovadora do computador pessoal, está em transformar cada consumidor num criador. Aproveite sua criatividade e envie suas colaborações, recebendo remuneração a título de DIREITO AUTORAL.

Se o material enviado for aprovado para publicação, a remuneração será de Cr$ 10.000,00 por programa. Se o programa for muito bom e muito extenso esta quantia será aumentada, a critério da redação, até Cr$ 30.000,00. Esta remuneração será oferecida também para artigos interessantes sobre o mundo do TK.

A maneira ideal de nos enviar o material a ser publicado obedece às seguintes normas:

1) **Nunca** esquecer de colocar nome completo, endereço e número de sua assinatura em **todo** material enviado (fitas, listagens de impressora, envelope, carta, etc).
2) Enviar a listagem de programa **datilografada** ou, melhor ainda, tirada na impressora do TK.
3) Colocar sempre uma linha **REM** com o nome do autor e o título do programa.
4) Enviar uma fita com o programa gravado **algumas vezes** (se possível em gravadores diferentes).
5) Na fita, gravar **com microfone** (em viva voz), algumas instruções úteis, **nome completo e endereço do remetente.**

O material não utilizado não será devolvido. Esta é uma norma corrente em quase todas as revistas do mundo, que gera porém, uma certa desconfiança: "e se apagarem meu nome do programa e o publicarem sem me remunerar?"

Neste ponto podemos sugerir o seguinte:

Ao enviar o material para **qualquer** revista, você, ou confia em sua seriedade, ou, é melhor não enviar o material, reservando-o para uma publicação dígna de seu crédito.

Uma outra saída é a de tirar uma cópia de todo material enviado e registrá-lo em cartório, para qualquer reclamação posterior. Obviamente tal procedimento só é aconselhável se o material enviado for valioso a ponto de justificar toda essa mão-de-obra.

COMO COLABORAR COM MICROHOBBY

Continuação...

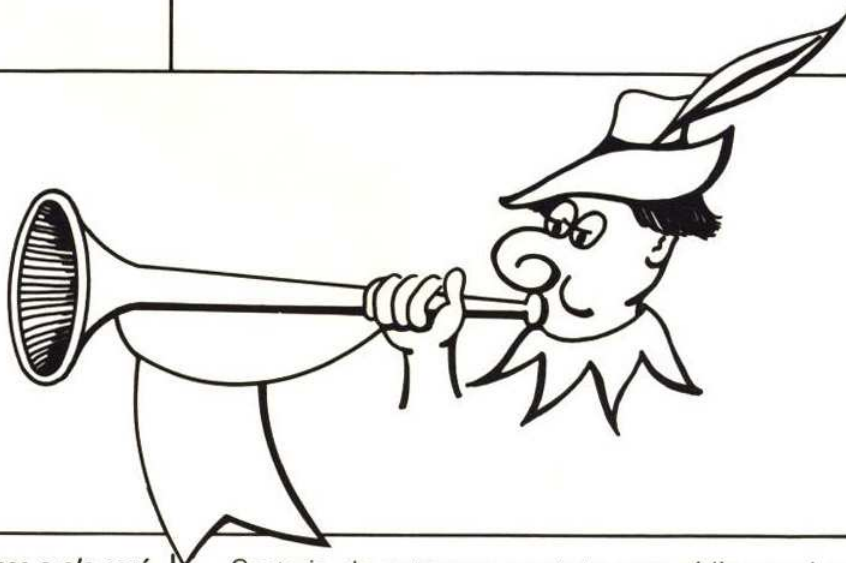
6) O material deve ser enviado para:

MICROMEGA P.M.D. Ltda.
PROGRAMAS DO LEITOR
Cx. POSTAL: 60081
CEP: 05096 — S. PAULO — SP

7) Qualquer dúvida, poderá ser esclarecida pelo telefone:

(011) 257-5767

Aguardamos sua colaboração.

PEQUENOS ANÚNCIOS

Envie seu pequeno anúncio de poucas linhas e ele será publicado gratuitamente pela MICROHOBBY.

* Vendo expansão de 16K para NEZ 8000, nenhum uso, baratíssima — José Luis — S. Paulo — Fone: 274-5995.

* SINCLAIR CLUB, para usuários do ZX81 — NEZ8000 — TK82-C — CP200 — TK85 — Envie uma carta com dados pessoais e um programa: Rua Bandeira Paulista, 147/162 — Itaim Bibi — São Paulo — CEP 04532.

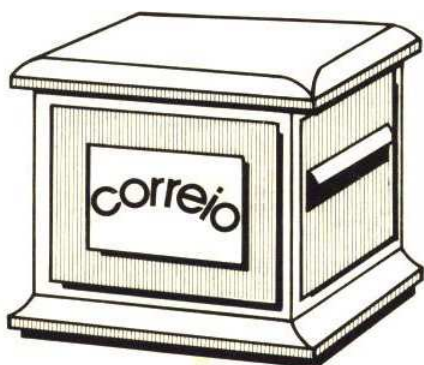
* Troca de idéias e programas para TK — Robson Gomes Vilela — Rua dos Navegantes, 1017/202 — Boa Viagem — Recife — PE — CEP 50000.

* Vendo programas em fita-cassete 2K e 16K para TK82 — Carlos Sciarretti — Caixa Postal 5567 — São Paulo — CEP 01051.

* Vendo programas em fita K-7 para os micros TK-82, NZ-8000, CP-200 e ZX-81. Aos interessados escrever para Renato Strauss, R. Cardoso de Almeida, 654 — ap. 32 — CEP 05013 — São Paulo — SP.

* Curso de linguagem de máquina para TK82 e TK85 (Assembly Z80). Aos sábados das 9:00 às 13:00 hs. — São Paulo — Fone: 813-4555 (Márcia) à tarde.

* Gostaria de entrar em contato com rádio amadores que utilizem micros do tipo Sinclair para decodificar CW e RTTY. Aos interessados entrar em contato com Renato Strauss, R. Cardoso de Almeida, 654 — ap. 32 — CEP 05013 — São Paulo — SP. Para trocas de idéias e experiências.

COMO FAZER SUA ASSINATURA

A nossa revista não será distribuída nas bancas. Para obter seu exemplar mensal, contendo muitos programas para o seu TK, muitas dicas e prêmios interessantíssimos, você deverá fazer uma assinatura: o preço anual da assinatura é de Cr$ 11.800,00. Porém, até 30 de setembro, manteremos o preço de Cr$ 9.900,00, com direito à escolha de um desses dois brindes:

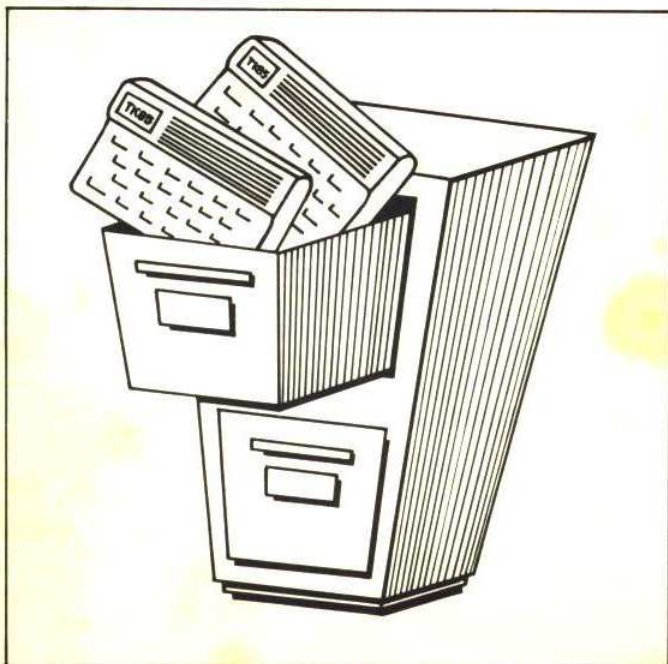
Fita com jogos (Pulga — 2K e Simulador de Vôo — 16 K) no valor de Cr$ 6.900,00.

TWIN-GO (Desgravador e rejuvenescedor instantâneo de fitas cassete) no valor de Cr$ 7.800,00

Para tanto, você deverá preencher corretamente o cupom anexo (não esquecendo de assinalar o brinde de sua escolha), e colocá-lo num envelope, junto a um cheque nominal ou vale postal a favor de MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO LTDA., no valor de Cr$ 9.900,00 (oferta válida até 30 de setembro).

O envelope deverá ser selado e endereçado à

MICROMEGA P.M.D. LTDA.
Caixa Postal 60081 — CEP 05096
São Paulo — SP

OLHE SÓ O QUE ESTÁ RESERVADO PARA O N.º 3

PROGRAMAS DO LEITOR:
Muitos programas interessantíssimos bolados ou adaptados por nossos leitores, inclusive aplicativos sérios.

PROGRAMAS DO MÊS:
Um gigantesco programa de arquivo.
O programa "LIFE" em linguagem de máquina.

DICAS:
Aritmética financeira.
Como preparar uma linha REM para código de máquina.
Como encher a tela instantaneamente.

CURSOS:
Continuação do Curso Basic TK e do Assembly Z80 do TK.

PEQUENOS ANÚNCIOS:
Para esta seção você poderá enviar um pequeno anúncio para publicação gratuita.

FITA DO MÊS:
As instruções e as dicas do Simulador de Vôo.

PREPARE-SE PARA O

VESTIBULAR

DE UMA MANEIRA

DIFERENTE

No **NÚCLEO** você pode se inscrever escolhendo as matérias, dependendo de suas reais necessidades, aproveitando ao máximo seu tempo.
Se o seu problema é MATEMÁTICA, FÍSICA, QUÍMICA, BIOLOGIA, GRAMÁTICA, REDAÇÃO, FRANCÊS, ETC., temos os melhores professores dos melhores cursinhos de São Paulo, para lhe dar um atendimento INDIVIDUALIZADO.

TURMAS DE 18 ALUNOS!

Informações e inscrições para turmas de AGOSTO

Núcleo de orientação de estudos
Av. Brig. Faria Lima, 1451 - 3º - Cj. 31
Tel.: 813-4555 - CEP 01451 - São Paulo - SP

Desgravador e rejuvenescedor instantâneo de fitas K-7

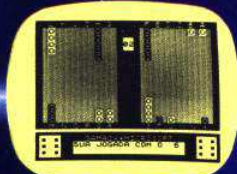
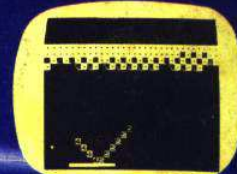
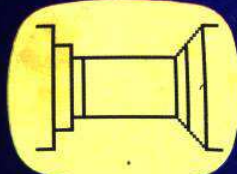
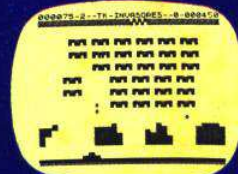
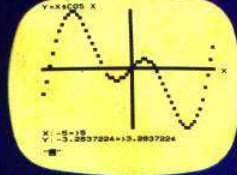
PREÇO: Cr$ 7.800,00

Sistema: PASSOU... DESGRAVOU
GARANTIA DE 5 ANOS

- Desgrava e rejuvenesce em apenas 4 segundos, qualquer fita cassete independente de sua duração, eliminando todos os ruídos e chiados decorrentes de frequentes regravações.
- Poupa o desgaste da fita e do gravador.
- Poupa tempo e melhora a qualidade do som da fita cassete na reutilização da mesma.
- Gravando em cima de outra gravação, a qualidade da reprodução é consideravelmente prejudicada, além de criar confusões, principalmente se a gravação anterior for de duração mair curta. Desgravando com o **TWIN-GO,** nada disso acontece.
- O **TWIN-GO** é indispensável a todos os profissionais e amadores, usuários de fitas cassete.

PARA APAGAR BASTA PASSAR A FITA IDA E VOLTA PELA ABERTURA

micromega

Pedidos por carta à
MICROMEGA PMD LTDA.
Caixa Postal 60081 - CEP 05096 - São Paulo-SP
anexando Cheque Nominal ou Vale Postal, incluindo mais despesas de frete no valor de Cr$ 500,00, autorizando o recebimento da compra no prazo de 15 dias.

MICROSOFT
Programas para o seu TK82-C e TK85
JOGO DE GAMÃO
16K
Este programa apresenta o tabuleiro no video e utiliza o eficiente codigo de maquina, permitindo 4 (quatro) niveis de dificuldades de jogo.
MONSTRO DAS TREVAS TRIDIMENSIONAL - 16K
Impressionante jogo onde você deve evitar o monstro das trevas. Tudo em 3 dimensões.
DEMOLIDOR
2K
Jogo animado, tipo "fliperama". O jogador devera demolir uma parede com uma bola que se encontra sempre em movimento.
LABIRINTO TRIDIMENSIONAL - 16K
Jogo em três dimensões. O jogador podera definir a dificuldade do Labirinto. O programa apresenta a posição do jogador em perspectiva. Em qualquer momento é possivel pedir auxilio ao computador.
INVASORES DO ESPAÇO - 16K
Consiste de uma frota de naves invasoras extraterrenas, descendo no planeta Terra. Sua missão é destruir as naves invasoras dispondo da arma de raios laser.
RALLY
16K
Emocionante corrida de rally em um labirinto, onde poderão ser testados sua habilidade e seus reflexos. Para conseguir seu intento, você deverá evitar carros-ataque e obstaculos em seu trajeto.
MATEMATICA I
16K/64K
Analise grafica de funções matematicas, resolução de sistemas de equações lineares (16K-51 equações/64K-95 equações), e Calculo de integrais definidos.
TK-MAN
16K
Jogo animado onde deverão ser apagados todos os pontinhos espalhados em um labirinto (o programa contém 15 tipos de labirinto). Você sera impedido a qualquer custo, por 4 extraterrenos, guardiães do labirinto, que poderão ser combatidos com cargas de raios-laser.
T-KALC
16K/64K
Programa desenvolvido para calculos numericos em planilha. O usuario define as colunas, as linhas e as formulas aplicadas. Similar ao famoso Visicalc. De grande versatilidade, este programa permite a formulação de calculos cientificos e comerciais, analise de tabelas numericas e outras aplicações.
TKADREZ II
16K
Este jogo apresenta o tabuleiro e as peças no video. Permite a escolha de ate 7 niveis de dificuldade. O programa fornece a qualquer momento, a listagem dos lances efetuados, e armazena em fita a posição das peças. Ele podera recomendar a sua jogada.
NOS REVENDEDORES AUTORIZADOS EM TODO PAÍS
80 programas à sua disposição. Solicite folheto.
MICROSOFT®
Av. Angélica 2.318 - 13º - Cj. 132
CEP 01228 - São Paulo - SP